KCB-Net: A 3D Knee Cartilage and Bone Segmentation Network via Sparse Annotation

SUBMISSION DATE / POSTED DATE

13-09-2021 / 16-09-2021

CITATION

Peng, Yaopeng; Zheng, Hao; Zaman, Fahim; Zhang, Lichun; Wu, Xiaodong; Sonka, Milan; et al. (2021): KCB-Net: A 3D Knee Cartilage and Bone Segmentation Network via Sparse Annotation. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.16611568.v1

DOI

10.36227/techrxiv.16611568.v1

KCB-Net: A 3D Knee Cartilage and Bone Segmentation Network via Sparse Annotation

Yaopeng Peng, Hao Zheng, Fahim Zaman, Lichun Zhang, Xiaodong Wu, *Senior Member, IEEE*, Milan Sonka, *Fellow, IEEE*, Danny Z. Chen, *Fellow, IEEE*

***Abstract*— Knee cartilage and bone segmentation is critical for physicians to analyze and diagnose articular damage and knee osteoarthritis (OA). Deep learning (DL) methods for medical image segmentation have largely outperformed traditional methods, but they often need large amounts of annotated data for model training, which is very costly and time-consuming for medical experts, especially on 3D images. In this paper, we report a new knee cartilage and bone segmentation framework, KCB-Net, for 3D MR images based on sparse annotation. KCB-Net selects a small subset of slices from 3D images for annotation, and seeks to bridge the performance gap between sparse annotation and full annotation. Specifically, it first identifies a subset of the most effective and representative slices with an unsupervised scheme; it then trains an ensemble model using the annotated slices; next, it self-trains the model using 3D images containing pseudo-labels generated by the ensemble method and improved by a bi-directional hierarchical earth mover's distance (bi-HEMD) algorithm; finally, it fine-tunes the segmentation results using the primal-dual Internal Point Method (IPM). Experiments on two 3D MR knee joint datasets (the Iowa dataset and iMorphics dataset) show that our new framework outperforms state-of-the-art methods on full annotation, and yields high quality results even for annotation ratios as low as 5%.**

***Index Terms*— Knee cartilage and bone segmentation; Sparse annotation; Ensemble learning; 3D MR images.**

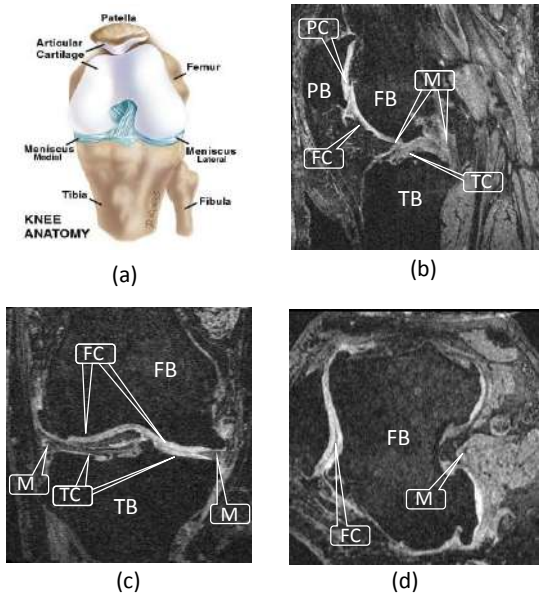


Fig. 1: Knee joint. (a) Anatomy of the knee joint (adopted from [1]). (b)-(d) Sagittal, coronal, and transverse MR image planes, showing the femur bone (FB), femoral cartilage (FC), tibia bone (TB), tibial cartilage (TC), patella bone (PB), patellar cartilage (PC), and meniscus (M).

I. Introduction

Osteoarthritis (OA) is a prevalent chronic disease caused by the damage and degeneration of cartilages. It is estimated that 20% of Americans may suffer from various levels of OA by 2030. Magnetic resonance imaging (MRI) has become a common technique for studying and assessing changes within the knee joint, including cartilages and bones. Fig. 1 illustrates the anatomical structure of the knee joint.

Considering the knee joint anatomy, the femoral cartilage (FC), tibial cartilage (TC), patellar cartilage (PC), and menisci (M) are the main tissues affecting the knee joint health. To quantitatively measure the thickness of the knee cartilages and identify the bone-cartilage interface, accurate cartilage and bone segmentation is needed.

To capture the detailed structure of the knee anatomy, 3D MR images are commonly scanned at high in-plane resolution. However, labeling 3D MR images is very time-consuming.

The best current methods for knee-joint segmentation, as to be discussed in Section II, depend on large-sized training data to learn segmentation parameters. But, forming large enough annotated datasets is difficult in medical image analysis. In this paper, we propose a new framework, KCB-Net, for 3D cartilage and bone segmentation with sparse annotation, and demonstrate its performance on a knee-joint segmentation task.

II. Related Work

Automated and semi-automated methods for knee joint segmentation have been investigated for several decades. Shape models, graph optimization approaches, and deep learning (DL) methods exhibited high performance in recent years.

Yaopeng Peng, Hao Zheng, and Danny Z. Chen are with the Dept. of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA (e-mails: {ypeng4, hzheng3, dchen}@nd.edu).

Fahim Zaman, Lichun Zhang, Xiaodong Wu, and Milan Sonka are with the Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242, USA (e-mails: {fahim-zaman, lichun-zhang, xiaodong-wu, milan-sonka}@uiowa.edu).

This research was supported in part by NIH NIBIB Grant R01-EB004640 and NSF Grant CCF-1617735.

3D graph based methods are well suited for knee cartilage segmentation. Yin et al. [2] proposed a layered optimal graph image segmentation for multiple objects and surfaces (LOGISMOS) framework to simultaneously segment multiple interacting surfaces of objects by incorporating multiple spatial interrelationships of surfaces in a D-dimensional graph. Kashyap et al. [3] extended the LOGISMOS framework to simultaneously segment 3D knee objects for multiple follow-up visits of the same patient – effectively performing optimal 4D (3D+time) segmentation. Xie et al. [4] proposed a primal-dual Internal Point Method (IPM) to first learn the parameters of the surface cost functions for the LOGISMOS algorithm and then solve an optimization problem for the final segmentation.

Several deep convolutional neural network (CNN) approaches showed close-to-human level performance. Liu et al. [5] proposed a fully automatic musculoskeletal tissue segmentation method that integrates CNN and 3D simplex deformable approaches to improve the accuracy and efficiency. Ambellan et al. [6] combined the strengths of statistical shape models and CNN to successfully segment knee bones/cartilages. Tan et al. [7] proposed a method to first extract the regions of interest (ROIs) for three cartilage areas and then fuse the three ROIs to generate fine-grained segmentation results.

Zheng et al. [8] proposed a 3D segmentation method that ensembles three 2D models and one 3D model (called base-learners). It first trains the base-learners using labeled data, and ensembles the base-learners by training a meta-learner [9]. It then re-trains the base-learners and meta-learner with pseudo-labels to obtain a 3D segmentation model. However, such base-learners still rely on fully annotated 3D data. In [10], Zheng et al. proposed a sparse annotation strategy to select the most representative 2D slices for annotation. It first encodes each slice into a low-dimensional vector, and prioritizes the slices based on their representativeness in a set of 3D images. Next, three 2D modules and one 3D module (3D FCN [11]) are trained, and pseudo-labels of the unlabeled data are generated using the base-learners. A Y-shape DenseVoxNet [9] is used to train a meta-learner, which ensembles the 2D and 3D modules. Zheng et al. [12] further extended this sparse annotation strategy, and designed a K-head FCN to compute the pseudo-label uncertainty of each slice and rule out highly uncertain pixels in the subsequent training process.

III. Method

A. Overview

Our KCB-Net combines and extends previously reported ensemble learning [8] and sparse annotation [10] methods for 3D segmentation. Fig. 2 shows its main steps. (1) *Representative slice selection*: As in [10], each 2D slice in every major *xy*, *yz*, or *xz* orientation in the entire set W of 3D training images is encoded as a low-dimensional latent vector, and all slices are prioritized by their representativeness. The top-ranked k slices are selected as the ones, in which to perform expert annotations. (2) *Base-learner training and pseudo-label generation*: As in [8], three 2D modules, one for each *xy*, *yz*, or *xz* orientations, are trained on the selected and annotated slices. Once 2D modules are trained, pseudo-labels are assigned to all remaining un-annotated slices in W and a 3D module is trained. KU-Net mechanism [13] is newly used to extract multi-scale features. Each module extracts information across different scales to support fine-scale feature extraction. Instead of using a sparse 3D FCN [11] as in [10], a DenseVoxNet [9] uses labels of the expert-annotated slices and pseudo-labels of the un-annotated slices. As in [14], an edge-aware branch is added to the 3D module to increase the weights of cartilage and bone surface locations. To explore the appearance consistency among consecutive slices and further improve the quality of the pseudo-labels generated, the H-EMD method [15] is newly enhanced by incorporating a bi-directional hierarchical earth mover's distance (bi-HEMD) when generating pseudo-labels of the un-annotated slices. Our bi-HEMD method first produces object candidates by applying multiple threshold values on the probability maps, and then selects object instances by minimizing the earth mover's distance based on a reference set of the object instances. (3) *Ensembling and self-training*: Following the pseudo-label generation, 2D and 3D modules are ensembled by training a 3D Y-shape DenseVoxNet [8] as a meta-learner using the original input images and pseudo-labels, which learns the target object segmentation from the labels/pseudo-labels. The output of the ensemble model is utilized to iteratively re-train the modules in Step (2) and the ensemble model in Step (3), repeated until convergence. (4) *Post-processing*: We newly add a post-processing step exploiting the task-specific characteristics that knee bones and cartilages are anatomically adjacent with one other. A fine-tuning network [4] that incorporates the surface interrelationships between adjacent bones and cartilages is trained by taking the probability maps generated in Step (3) as input and the pseudo-labels as the learning targets. The fine-tuning network is optimized using the IPM algorithm [4].

B. Representative Slice Selection

Identifying a small-enough set of the most representative 2D slices for annotation that subsequently facilitates the segmentation method training is critical for the success of our proposed approach. This section presents our slice selection scheme, called representative annotation (RA).

Medical experts often annotate a 3D image by choosing one orthogonal plane (*xy*, *yz*, or *xz*) and labeling the corresponding slices one by one. It may, however, be beneficial to annotate 2D slices along each of the three orthogonal planes. Fig. 3 illustrates the slice selection method.

1) Slice Representation: For a specified annotation ratio (e.g., 10% of all slices), to select the most representative slices to label, we first need to efficiently represent the slices. Medical image slices can commonly be represented as latent feature vectors of a much smaller size compared to the original 2D image matrix. By comparing slices using their latent vectors, not only can we reduce the computation cost but also extract their most useful information.

We utilize an auto-encoder as the representation extractor for the slices in our 3D training image set W, which learns efficient features in an unsupervised manner and conducts a lossy compression in the encoding process. It learns to store relevant information and disregard noise. This auto-encoder

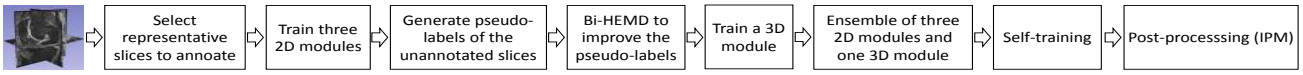


Fig. 2: The pipeline of our proposed KCB-Net framework.

consists of two parts: An encoder produces a compressed knowledge representation x for an input image (or slice) I; a decoder takes the representation x as input and outputs $\hat{x}$ as a reconstruction of the original image. The entire auto-encoder model is optimized by minimizing the sum of the reconstruction error $\mathcal{L}(x, \hat{x})$, which measures the differences between the original image and the reconstruction produced, and a regularization term for alleviating overfitting. This can be formulated as:

$$\phi^*, \psi^* = \arg\min_{\phi,\psi}(\mathcal{L}(x, \hat{x}) + \lambda_1 \times \sum_{i=1}^{M} w_i^2), \quad (1)$$

where $\mathcal{L}$ is the reconstruction loss between x and $\hat{x}$, λ_1 is a scaling parameter for the regularization term $\sum_{i=1}^{M} w_i^2$ to adjust the trade-off between the sensitivity to the input and overfitting, w_i is the i-th parameter of the auto-encoder, and ϕ and ψ are the parameters of the encoder and decoder, respectively.

To facilitate a fast training and convergence of the auto-encoder, we use a ResNet-101 [16] pre-trained on ImageNet [17] as the encoder backbone. A light-weight decoder (ResNet-50 [16]) is added to map the latent vectors to the original input space. Since slices along each orthogonal plane will be selected, we train the auto-encoder using all the slices of the 3D training set W along the three orthogonal planes.

2) Prioritizing the Slices: After training the auto-encoder, we measure the representativeness of each slice in the 3D training image set W as in [10]. First, we feed a 2D slice I to the encoder, and take the generated latent vector f as the representation of the slice I. Second, we define and compute the similarity between two slices I_i and I_j as $Sim(I_i, I_j) = cosine(f_i, f_j)$, where f_i and f_j are the latent vectors of I_i and I_j respectively, and $cosine$ denotes cosine similarity.

Next, a subset S of slices is selected from all the slices $S(W)$ of the set W (for an annotation ratio or a given size of S). The representativeness of S with respect to W is defined as:

$$F(S, W) = \sum_{I \in S(W)} \max_{I_s \in S}(Sim(I_s, I)). \quad (2)$$

Finding an optimal slice subset S was formulated as a maximum cover problem in [10], which is NP-hard, and a polynomial time approximation solution was obtained using a greedy method. Suppose a subset S' is the most representative for the images in W. The next choice (if needed) is a slice I^* in the remaining slice set $S(W) - S'$ that maximally increases the representativeness of the new subset $S' \cup \{I^*\}$, i.e.,

$$I^* = \arg \max_{I \in (S(W) - S')}(F(S' \cup \{I\}, W) - F(S', W)). \quad (3)$$

This selection process puts all the slices in W in decreasing order based on their representativeness. The slices with better representativeness have higher priorities for annotation.

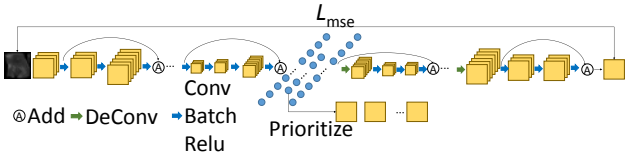


Fig. 3: Illustrating the representative slice selection method. L_{mse} denotes the mean square error.

C. Base-learner Training and Pseudo-label Generation

After the representative slice selection, the selected slices are labeled by experts, which we denote as $S_L = \{S_{l_1}, S_{l_2}, \ldots, S_{l_N}\}$, where l_N is the number of slices selected. Due to the limited training data, we apply the bottleneck structure in [18], which can achieve better performance compared to a common U-Net since it has fewer parameters and thus can alleviate overfitting.

To better exploit multi-scale features of the objects in our 3D knee images, we apply the KU-Net design in [13] to our backbone network to build a K-FCN network, which consists of K FCN submodules connected sequentially as in [13]. K-FCN first extracts information at different scales sequentially and then feeds the extracted information to the subsequent FCN submodules to assist feature extraction in finer scales. We apply the FCN structure in [18] as the backbone (with fewer parameters than U-Net). The first submodule of K-FCN is used to extract coarser-scale features, which are fed to the next submodule to extract features in a finer scale. The structure of our K-FCN is shown in Fig. 4 (with $K = 2$).

A 2D segmentation model can use a relatively large receptive field, but it does not utilize the interactions between consecutive slices well, which may result in spatial slice-to-slice inconsistency. Hence, we follow the ensemble method in [8] and train a 3D module, which produces smoother 3D results. We choose DenseVoxNet [9] as the backbone for our 3D module, since it has better parameter efficiency and thus a smaller chance to incur overfitting, especially with limited training data. Likewise, we use the KU-Net design and build a K-DenseVoxNet to exploit 3D multi-scale features. The coarse features extracted by the first DenseVoxNet submodule are fed to the second submodule to obtain fine-grained features.

For knee joint segmentation, the bone and cartilage boundaries are more important than other areas, since they usually serve as the main criteria to measure whether a cartilage is damaged. Hence, adding an edge-aware regulation can force the network to focus more on the boundary areas. Fig. 5 shows the structure of our edge-aware K-DenseVoxNet. The edge gate $F_{L\rho G}$ is defined as:

$$F_{L\rho G}(I) = k_G * \rho(k_L * I), \quad (4)$$

where k_G and k_L represent the Gaussian smoothing kernel and Laplacian kernel respectively, $*$ denotes convolution, and

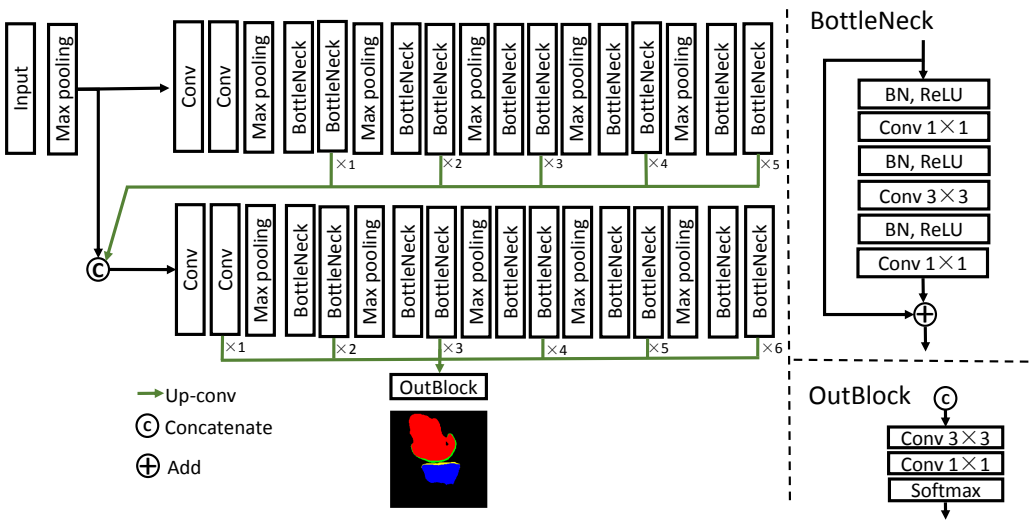


Fig. 4: The structure of our K-FCN ($K = 2$).

ρ is an activation function.

The loss function of our 3D module is defined as:

$$\mathcal{L} = L_{region} + \lambda_2 L_{edge}, \tag{5}$$

where L_{region} and L_{edge} are the cross entropy losses of the region branch and edge branch respectively, and λ_2 is a scaling parameter to regularize the edge branch.

We first train our three 2D segmentation modules using the selected labeled slices for each of the three orthogonal planes, and generate the probability maps of the unlabeled slices using the three trained 2D modules. We then train our 3D edge-aware K-DenseVoxNet using the 3D images in W that contain both the labeled slices and unlabeled slices that are now "labeled". Specifically, the pseudo-labels produced by the three 2D modules are first improved by the bi-HEMD algorithm in Section III-D. Then, the probability maps attained by the three 2D modules are averaged to generate the pseudo-labels used for training our 3D module. These four trained segmentation modules generate their pseudo-labels respectively for all the unlabeled slices. For simplicity, we average the results of these four modules as the probability map for each 3D image in W.

D. Bi-directional Hierarchical Earth Mover's Distance

After training our three 2D modules, probability maps of all the unlabeled slices in W are obtained. One observation on the 3D knee images is that the appearances of bones and cartilages between consecutive slices are often similar in size and shape. Exploring such appearance similarity can help improve the pseudo-label quality. Hence, we apply the hierarchical earth mover's distance (H-EMD) method [15] that uses many threshold values of the probability map for each unannotated slice and exploits the appearance consistency between consecutive slices to optimize the pseudo-labels.

The H-EMD method [15] takes two key steps. (i) Candidate instance generation: For a set of v threshold values, $\{t_h\}_{h=1}^{v}$, from the probability map of a slice S_i in a 3D image, produce a set IC_i of possible object instance candidates. These object candidates can be organized into a forest structure F_i. Also, a reference set R_{i-1} of object instances is built on the slice S_{i-1} (obtained iteratively). (ii) Candidate instance selection: For each pair of an instance candidate in F_i and a reference instance in R_{i-1}, compute their matching score as the cosine distance between their instance feature vectors. The goal is to maximize the sum of the weighted matching scores between the candidate set IC_i and reference set R_{i-1} to select the "best" object instances for the slice S_i. This can be solved by integer linear programming. For a dataset with n different classes, a feature vector for each instance candidate is defined as $(x, y, z, v_1, \ldots, v_n)$, whose first three items are the coordinates of its center pixel and the last n items are for an n-D one-hot vector denoting the category of the instance.

Rather than using the Euclidean distance as in [15], our method applies cosine distance, since our vectors contain two different types of information, which make the L_2 distance unsuitable to measure the differences between these vectors. Similar to bi-directional RNN [13], we perform the H-EMD process in two opposite directions (bi-HEMD). That is, for any two labeled slices S_i and S_j in a 3D image, $i < j$, we apply H-EMD along the direction of $S_{i+1}, S_{i+2}, \ldots, S_{j-1}$, and along $S_{j-1}, S_{j-2}, \ldots, S_{i+1}$. With the bi-HEMD process, the pseudo-labels generated by the 2D modules are improved, which are then used to train the 3D module in Section III-C.

E. Tuning the Final 3D Model Using Pseudo-labels

We now have three 2D K-FCNs and one 3D K-FCN trained with labeled or pseudo-labeled slices along the xy, yz, and xz planes. Next, we produce the probability maps of each 3D image M in W using these four FCN modules, denoted as m_{xy}, m_{yz}, m_{xz}, and m_{3D}, respectively. These probability maps are averaged, and the results are used to train our 3D meta-learner. This meta-learner is a Y-shaped K-DenseVoxNet that is aware of the raw images and their pseudo-labels so as to ease overfitting. Fig. 6 shows our meta-learner.

After training our 3D meta-learner, we apply the self-training strategy in [8] to further improve the model perfor-

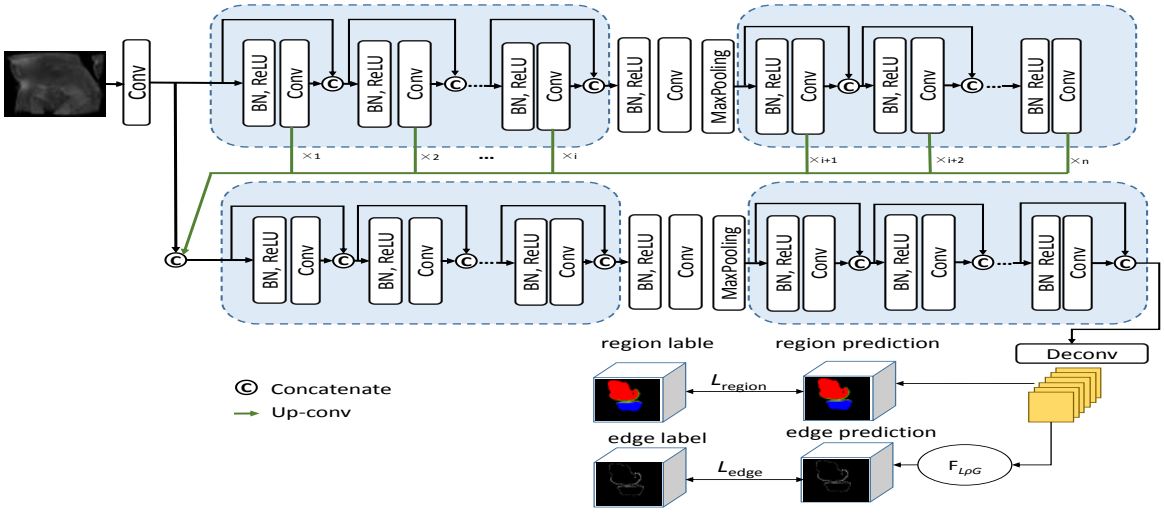


Fig. 5: The structure of our K-DenseVoxNet with edge-aware branches ($K = 2$).

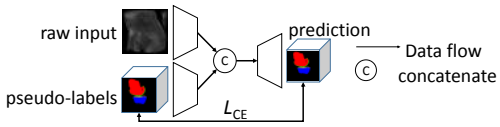


Fig. 6: The structure of our meta-learner.

mance. In this self-training process, the segmentation results of the meta-learner are regarded as pseudo "ground truth" of the unlabeled slices, which are used to re-train the 2D/3D base-learners (the three 2D base-learners are re-trained with the "labeled" slices along the three orthogonal planes). Note that the base-learners are first trained in the step of Section III-C. Here, we apply the SGD optimizer and a smaller learning rate to ensure the robustness and convergence of the entire training process. The loss function L_{CE} of the 3D meta-learner (see Fig. 6) is defined as the cross-entropy between the predictions and input pseudo-labels. The base-learners are re-trained, and generate four versions of pseudo-labels for each 3D image in W, which are averaged and used to train the meta-learner again. We repeat this self-training process for a few iterations, until the meta-learner performance no longer improves, giving rise to our final 3D model.

F. Post-processing Using IPM

Instead of applying the softmax function to the final probability maps, we further perform some post-processing to fine-tune the probability maps. One observation is that the surfaces of bones and cartilages are mutually "coupled" in some areas, within which the topology and relative positions of the bones and cartilages are known and the distances between them are within specific ranges. Furthermore, physicians care more about the "coupled" areas since osteoarthritis is usually caused by damages of the knee cartilages in such areas. Thus, we apply the IPM method [4] by incorporating the surface interrelationships between the bones and cartilages into the segmentation process to further improve the segmentation performance. An advantage of the IPM method over traditional graph based methods is that it parameterizes the surface cost functions in the graph model and leverages DL to learn the parameters rather than relying on hand-crafted features.

Instead of using ground truth to train the surface segmentation network of IPM [4], we use the pseudo-labels generated by our meta-learner to optimize this network in the first iteration. Afterwards, the pseudo-labels are updated by IPM and used to re-train the network. Such operations are repeated several times until convergence. The details of the above training process are shown in Fig. 8 [4].

Since the bone and cartilage surfaces are not terrain-like, we need to first unfold the knee joint into seven parts following the practice in [19], i.e., the front, back, top, center, bottom, left and right parts, respectively, as shown in Fig. 7.

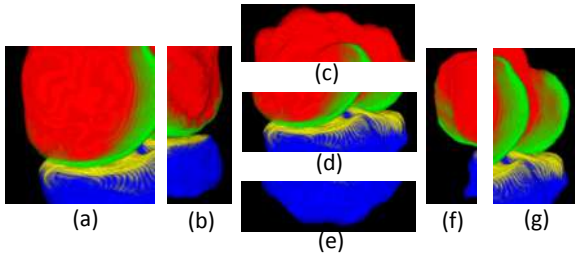


Fig. 7: Illustrating the seven unfolded parts of the knee joint. The corresponding parts in the sagittal view are: (a) front; (b) back; (c) top; (d) center; (e) bottom; (f) left; (g) right.

Specifically, for the center part (see Fig. 7(d)), we replace U-Net used in the original IPM method [4] with the probability maps generated by our final fine-tuned ensemble model. Finally, we patch its 6 junction areas (i.e., the junction areas between center and front, center and back, center and top, center and bottom, center and left, and center and right), and average the center area and its corresponding junction areas processed by IPM to smooth the final results.

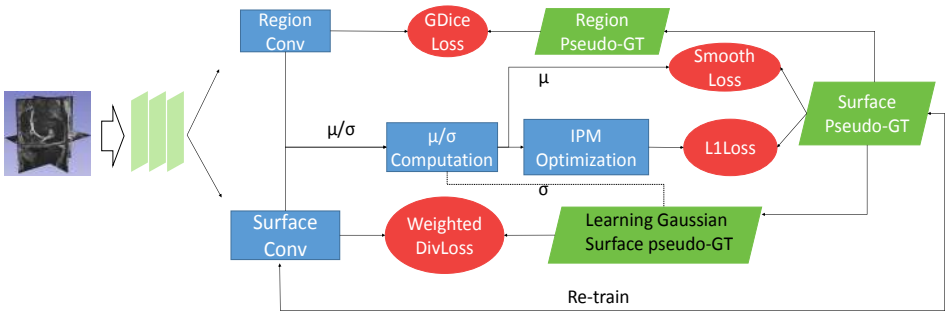


Fig. 8: The process of the post-processing step [4].

IV. Experiments and Analysis

To demonstrate the capabilities of our KCB-Net approach, its performance was compared with state-of-the-art knee segmentation methods using full annotations as well as compared with two state-of-the-art slice selection strategies: equal-interval annotation (EIA) and random slice selection (RSS). Furthermore, the effect of each component in our KCB-Net framework was assessed and the robustness of the method was quantified for different sparse annotation ratios.

A. Datasets and Implementation Details

The performance of our KCB-Net model was evaluated on knee joint images from the Osteoarthritis Initiative database (OAI, http://www.oai.ucsf.edu/). The image size is $384\times384\times160$, with voxel size of $0.36mm \times 0.36mm \times 0.7mm$. Two subsets with ground truth are available: (1) A University of Iowa annotated portion of the OAI that was first segmented by the LOGISMOS method [3] and the automated segmentations then corrected by the just-enough-interaction (JEI) approach in 4D (3D+time) [20]. This Iowa dataset consists of 1462 double echo steady state (DESS) 3D MR images from 248 subjects. Four compartments are annotated: femur bone (FB), femoral cartilage (FC), tibia bone (TB), and tibial cartilage (TC). (2) The iMorphics dataset, available directly from the OAI database, includes 176 3D MR knee images acquired with 3T Siemens MAGNETOM Trio scanners and quadrature transmit-receive knee coils (USA Instruments, Aurora, OH, USA). The annotated compartments are femoral cartilage (FC), tibia cartilage (TC), patellar cartilage (PC), and menisci (M).

We implemented all the networks using PyTorch [21]. For our auto-encoder, ResNet-101 [16] is used as the backbone of its encoder and ResNet-50 [16] as the backbone of its decoder. The encoder is initialized with a model pre-trained on ImageNet [17]. All the other parameters are initialized as in [16], and λ_1 in Eq. (1) is set to $5e$-5. The network was optimized using the Adam optimizer (learning rate = $1e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). The 3D images were first cropped so as to remove the background clearly outside of the knee area. Each slice or 3D image was normalized to zero mean and unit standard variance. In the data augmentation for 3D model training, starting points are randomly selected in a 3D image, and a patch of size $80 \times 192 \times 160$ is cropped at each starting point. Afterwards, common spatial transforms (e.g., rotation, scaling, and mirroring) are applied. In 2D model training, each slice is augmented with common spatial transforms.

We set $K = 2$ for the K-FCNs and K-DenseVoxNet with edge-aware branches (for larger K, the model costs increase largely but the accuracy improves little [13]). We use *mean square error* as the auto-encoder's loss. We set the parameter of the edge regularizer in the edge-aware K-DenseVoxNet as $\lambda_2 = 1e-4$ (see Eq. (5)).

B. Evaluation Metrics

Dice similarity coefficient (DSC) and average symmetric surface distance (ASSD, in mm) between the labeled and segmented surfaces are used as our evaluation metrics.

1) Dice Similarity Coefficient: Dice similarity coefficient (DSC) is calculated as:

$$DSC = \frac{2 \times V(GT \cap Pred)}{V(GT) + V(Pred)}, \tag{6}$$

where GT is the ground truth, $Pred$ is the prediction, and $V(X)$ denotes the volume of a 3D object X.

2) Average Symmetric Surface Distance: Average symmetric surface distance (ASSD) focuses on the absolute distances between surfaces of the segmented objects and their ground truths, calculated as:

$$ASSD = \frac{1}{n\partial A + n\partial B}\left(\sum_{a\in\partial A} d(a,\partial B) + \sum_{b\in\partial B} d(b,\partial A)\right), \tag{7}$$

where ∂A and ∂B denote the surfaces of objects A and B respectively, $n\partial A$ and $n\partial B$ denote the numbers of voxels on ∂A and ∂B respectively, and $d(x,\partial S)$ denotes the nearest Euclidean distance of a point x to a surface ∂S.

C. Experimental Results on Full Annotation

To obtain robust results, we conduct 5-fold cross validation on the Iowa and iMorphics knee datasets. For the Iowa dataset, 1170 3D images (out of 1462) are for training and 292 3D images for testing in each fold. For the iMorphics dataset, 140 3D images are for training and 36 3D images for testing.

Table I shows the performance comparison of KCB-Net and other methods trained on fully annotated Iowa dataset. The Iowa dataset was also used for other comparisons as follows: (i) 4D LOGISMOS [3]: utilizing a hierarchical set of random forest classifiers to learn the cartilage appearance

and simultaneously segment multiple interacting surfaces of objects based on an algorithmic incorporation of multiple spatial interrelationships in an n-dimensional graph. (ii) The ensemble learning method [10]: Ensembling four 2D/3D FCNs and self-training with fully labeled 3D data.

From Table I, one can see that our KCB-Net outperforms LOGISMOS-4D on both the bone and cartilage segmentations. KCB-Net also outperforms the ensemble method [10], which demonstrates that the KU-Net design, edge-aware DenseVoxNet, bi-HEMD method, and IPM post-processing method that we use help improve the segmentation performance.

Table II presents the results achieved on the fully annotated iMorphics dataset. We compare with three recent methods: (i) UDA [22]: utilizing mixup and adversarial unsupervised domain adaptation to improve the robustness of DL-based knee cartilage segmentation in new MRI acquisition settings; (ii) CML [7]: detecting the regions of interest and fusing the cartilages by a fusion layer; (iii) the ensemble method [10]. Our method attains better DSC scores on FC, TC, PC, and M compared to the UDA method. We also outperform the CML and ensemble methods in both DSC and surface errors of FC, TC, and PC, suggesting that our method can obtain more quantitatively accurate knee cartilage/bone segmentation.

Performance improvement of our new method over the original ensemble method [10] was evaluated on the Iowa and iMorphics datasets, using paired t-tests. Tables I and II show that in most compared cases, our new approach significantly outperforms the earlier approach [10] (with $p < 0.05$).

D. Experimental Results on Sparse Annotation

To evaluate the performance of our method on sparsely annotated data, we compare its performances on data with changing sparse annotation ratios vs. those achieved using different slice selection schemes. Specifically, we compare the representative annotation (RA) scheme used in our KCB-Net pipeline with two common slice selection schemes: equal-interval annotation (EIA) and random slice selection (RSS). Suppose for an annotation ratio, S_k slices are to be selected. The EIA scheme selects $S_k/3$ slices at equal distance along each axis, and the RSS scheme randomly selects $S_k/3$ slices along each axis. We repeat the RSS process 10 times, and take the average of the results as the RSS base performance. Figs. 9 and 10 show the performance comparison with various annotation ratios on the Iowa and iMorphics datasets, respectively.

From Figs. 9 and 10, one can see that our RA outperforms the EIA and RSS schemes on both the cartilage and bone segmentations. Our method can notably alleviate performance degradation, especially for annotation ratios $\leq 5\%$. This is because EIA selects the locationally same slice indices in each 3D image, which might make the trained model overfit on the selected slices and cause segmentation errors on the remaining slices. RSS performs better than EIA in very sparse annotation ratios ($< \%10$) but worse than EIA in less sparse annotation ratios ($> \%40$), since RSS can select different slices in different 3D images, likely incurring less overfitting.

Another observation from Figs. 9 and 10 is that the performance drops drastically when the annotation ratios are $< 5\%$, suggesting that this annotation ratio may be the "lower limit" for a satisfactory performance on knee segmentation.

To examine the statistical significance of the improvements of RA over EIA and RA over RSS, we computed the p-values for RA over EIA, and RA over RSS at different annotation ratios. We observed that the improvements of RA over EIA and RA over RSS are typically statistically significant (p-values < 0.05) when the annotation ratios are quite small ($\leq 20\%$); for larger annotation ratios ($> 20\%$), the p-values tend to be ≥ 0.05. We think the reason for this trend is that for dense annotations, the chance of selecting the same or similar slices by different selection schemes increases quickly. Figs. 11 and 12 illustrate this trend on the Iowa and iMorphics datasets using the range 0%–30% of annotation ratios.

E. Ablation Study

To examine the contribution of each component in our KCB-Net, we conducted the ablation study to compare the performances of its components, denoted as follows. (1) S1: 2D xy module; (2) S2: 2D yz module; (3) S3: 2D xz module; (4) S4: 3D module; (5) S5: ensembling of the three 2D modules and the 3D module; (6) S6: bi-HEMD; (7) S7: self-training; (8) S8: IPM post-processing.

Performance of each individual component in S1, S2, S3, and S4 is given first, followed by the ensemble performance (S5) that combines all these four components. For S6–S8, components were repeatedly added to the framework each time; the more the performance increases, the more important the corresponding component (in S6–S8) is. Thus, note that S8 actually reflects the performance of the entire framework including all its components.

Tables III and IV present the ablation study results on the Iowa and iMorphics datasets, respectively. We observe that the ensemble of the 2D and 3D modules can substantially improve the performance over the individual modules. The 3D module often attains better performance than the 2D modules since it exploits the inter-relations among consecutive slices. The ensemble strategy can benefit from both the 2D modules (with a large receptive field) and the 3D module (exploiting the interactions among consecutive slices). Since some cartilages are very thin along the sagittal plane, it is quite difficult for DL models to detect them along such a plane, especially with very sparse annotation. Utilizing other 2D modules can help address this issue. Both Table III and Table IV show that the ensemble strategy and the self-training mechanism play more important roles than the other components. Figs. 13 and 14 qualitatively compare results in the sagittal view on the Iowa and iMorphics datasets.

F. Discussion

From Figs. 9 and 10, one can see that our representative annotation (RA) scheme substantially reduces the performance gap between different annotation ratios, meaning that our framework can achieve comparatively good results while using much less annotated data than required for full annotation. Our ensemble method and the self-training using pseudo-labels improved by the bi-HEMD method largely improve the

TABLE I: Comparison with state-of-the-art methods using full annotation on the Iowa dataset. Here, "–" denotes that the corresponding results were not reported in the original paper. Paired t-test values indicate the significance status of the improved performance of our method vs. the ensemble method [10] (statistically significant improvements are in bold).

	Femur Bone		Femoral Cartilage		Tibia Bone		Tibial Cartilage	
	DSC	ASSD	DSC	ASSD	DSC	ASSD	DSC	ASSD
LOGISMOS-4D [3]	–	–	–	0.55±0.11	–	–	–	0.60±0.14
Ensemble method [10]	0.940±0.011	0.551±0.017	0.830±0.020	0.541±0.010	0.930±0.131	0.557±0.156	0.812±0.034	0.590±0.177
Our method	**0.961±0.006**	**0.515±0.020**	**0.835±0.027**	0.522±0.009	0.957±0.102	**0.521±0.143**	**0.817±0.039**	**0.565±0.132**
p-value	0.043	0.001	0.047	0.066	0.071	0.009	0.032	0.044

TABLE II: Comparison with state-of-the-art methods using full annotation on the iMorphics dataset. Here, "–" denotes that the corresponding results were not reported in the original paper. Paired t-test values indicate the significance status of the improved performance of our method vs. the ensemble method [10] (statistically significant improvements are in bold).

	Femoral Cartilage		Tibial Cartilage		Patellar Cartilage		Menisci	
	DSC	ASSD	DSC	ASSD	DSC	ASSD	DSC	ASSD
UDA [22]	0.907±0.019	–	0.897±0.028	–	0.871±0.046	–	0.863±0.034	–
CML [7]	0.900±0.037	–	0.889±0.038	–	0.880±0.043	–	–	–
Ensemble method [10]	0.908±0.019	0.218±0.054	0.903±0.030	0.187±0.065	0.887±0.018	0.360±0.422	0.880±0.021	0.305±0.221
Our method	**0.919±0.020**	0.212±0.096	**0.909±0.025**	**0.184±0.068**	**0.900±0.026**	0.348±0.409	**0.889±0.024**	**0.295±0.210**
p-value	$\ll 0.001$	0.233	0.001	0.002	$\ll 0.001$	0.312	$\ll 0.001$	0.002

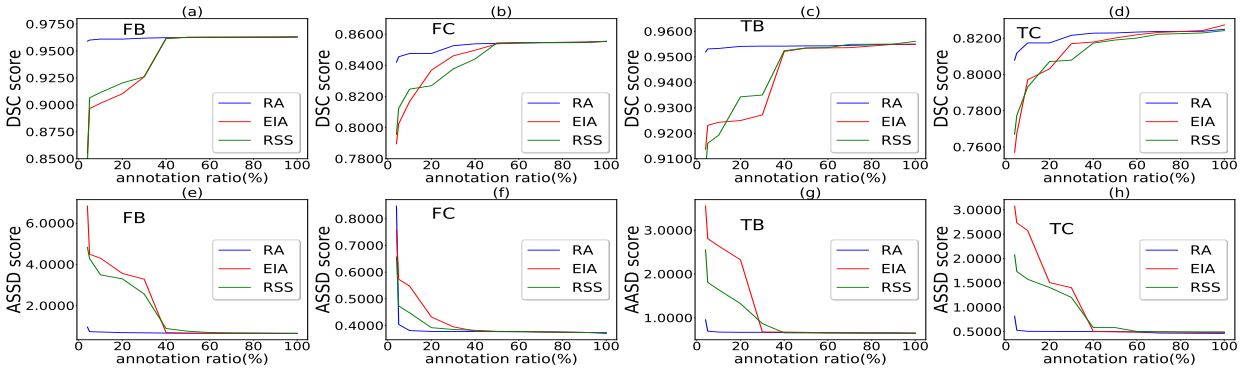


Fig. 9: Comparison of three slice selection schemes (RA, EIA, RSS) on the Iowa dataset.

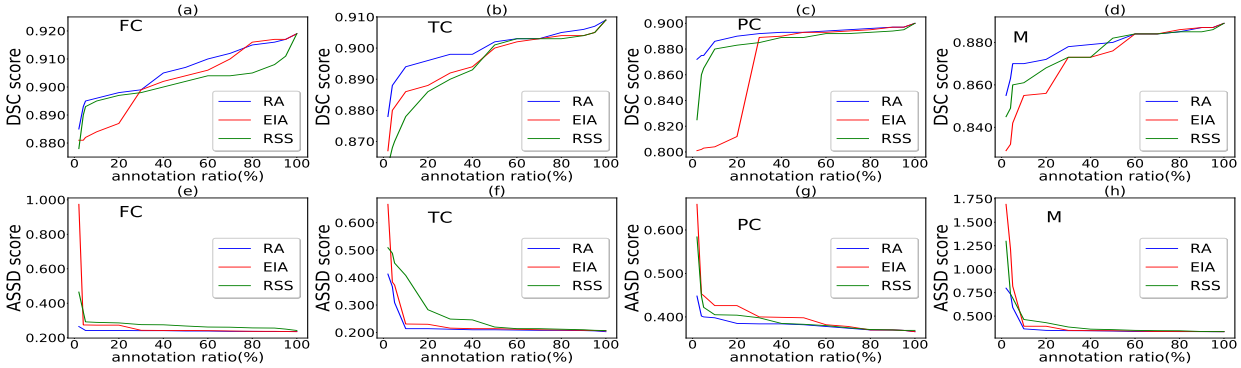


Fig. 10: Comparison of three slice selection schemes (RA, EIA, RSS) on the iMorphics dataset.

TABLE III: Ablation study of our method on the Iowa dataset.

	Femur Bone		Femoral Cartilage		Tibia Bone		Tibial Cartilage	
	DSC	ASSD	DSC	ASSD	DSC	ASSD	DSC	ASSD
S1 (xy)	0.938±0.025	0.550±0.027	0.815±0.037	0.557±0.015	0.924±0.187	0.562±0.109	0.796±0.040	0.603±0.192
S2 (yz)	0.931±0.016	0.562±0.020	0.811±0.025	0.566±0.016	0.916±0.129	0.573±0.217	0.790±0.086	0.599±0.210
S3 (xz)	0.936±0.019	0.558±0.023	0.812±0.024	0.564±0.009	0.918±0.163	0.571±0.156	0.792±0.069	0.602±0.191
S4 (3D)	0.940±0.023	0.550±0.016	0.817±0.012	0.556±0.011	0.926±0.125	0.560±0.147	0.796±0.094	0.601±0.221
S5 (ensemble)	0.947±0.007	0.540±0.014	0.820±0.039	0.552±0.019	0.933±0.109	0.552±0.233	0.804±0.033	0.613±0.219
S6 (bi-HEMD)	0.949±0.006	0.545±0.018	0.822±0.021	0.548±0.015	0.937±0.133	0.553±0.164	0.805±0.086	0.609±0.126
S7 (self-training)	0.957±0.007	0.517±0.012	0.831±0.035	0.528±0.010	0.950±0.082	0.524±0.143	0.814±0.036	0.572±0.138
S8 (IPM)	0.961±0.006	0.515±0.020	0.835±0.027	0.522±0.009	0.957±0.102	0.521±0.143	0.817±0.039	0.565±0.132

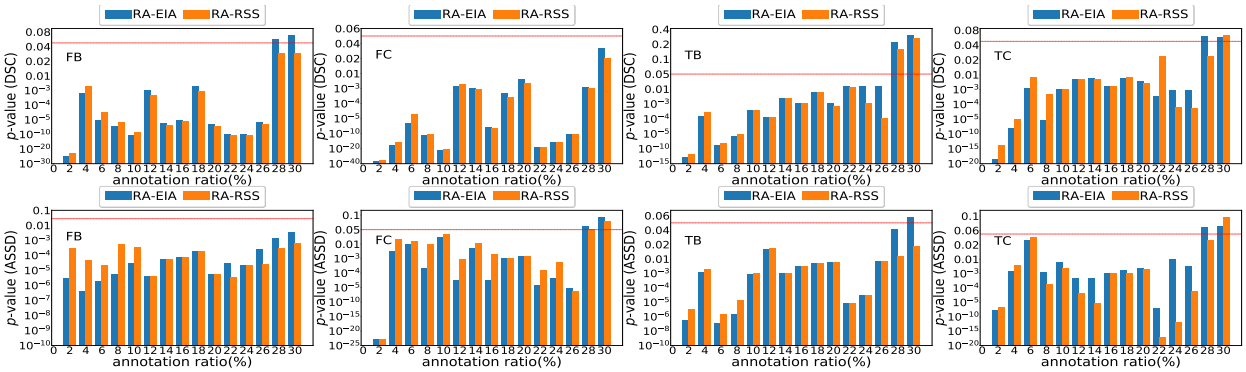


Fig. 11: Significance of performance improvements of employing our RA scheme vs. the EIA and RSS schemes on the Iowa dataset. The performance improvement is statistically significant if the charted p-value is below the red dashed line ($p < 0.05$). Experiments were performed in annotation ratio steps of 2%. To allow the very small (highly significant) p-values (e.g., $p < 0.01$) to be visible, the y-axes are piece-wisely adjusted and labeled to help improve the readability.

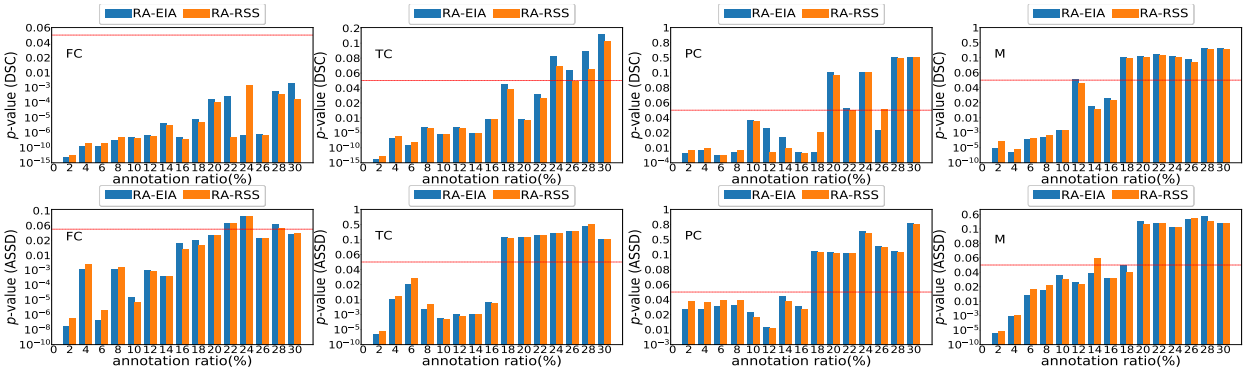


Fig. 12: Significance of performance improvements of employing our RA scheme vs. the EIA and RSS schemes on the iMorphics dataset. The performance improvement is statistically significant if the charted p-value is below the red dashed line ($p < 0.05$). Experiments were performed in annotation ratio steps of 2%. To allow the very small (highly significant) p-values (e.g., $p < 0.01$) to be visible, the y-axes are piece-wisely adjusted and labeled to help improve the readability.

TABLE IV: Ablation study of our method on the iMorphics dataset.

	Femoral Cartilage		Tibial Cartilage		Patellar Bone		Menisci	
	DSC	ASSD	DSC	ASSD	DSC	ASSD	DSC	ASSD
S1 (xy)	0.890±0.022	0.251±0.051	0.876±0.021	0.227±0.074	0.854±0.072	0.378±0.439	0.847±0.025	0.348±0.086
S2 (yz)	0.885±0.020	0.258±0.045	0.873±0.022	0.231±0.048	0.848±0.060	0.382±0.107	0.850±0.027	0.352±0.077
S3 (xz)	0.889±0.020	0.252±0.071	0.873±0.021	0.230±0.045	0.850±0.168	0.382±0.280	0.848±0.023	0.351±0.207
S4 (3D)	0.891±0.020	0.250±0.055	0.877±0.025	0.224±0.067	0.854±0.157	0.376±0.240	0.851±0.024	0.343±0.100
S5 (ensemble)	0.901±0.022	0.238±0.050	0.882±0.048	0.219±0.068	0.871±0.155	0.362±0.252	0.858±0.025	0.314±0.200
S6 (bi-HEMD)	0.902±0.020	0.236±0.082	0.887±0.020	0.213±0.072	0.877±0.051	0.360±0.402	0.864±0.024	0.317±0.201
S7 (self-training)	0.913±0.022	0.215±0.050	0.905±0.024	0.189±0.065	0.896±0.070	0.348±0.033	0.886±0.027	0.297±0.155
S8 (IPM)	0.919±0.020	0.212±0.096	0.909±0.025	0.184±0.068	0.900±0.026	0.348±0.409	0.889±0.024	0.295±0.210

segmentation performance, because the training data we use contribute new information in a more efficient way. Figs. 13 and 14 show that our ensemble and self-training strategies allow detection of small objects and thin boundary areas, despite the annotation sparsity. Our IPM post-processing helps further fine-tune the boundary areas, making the segmentation results more accurate and reliable overall.

V. Conclusions

We reported a new framework, KCB-Net, for segmenting cartilage and bone surfaces in 3D knee joint MR images. Our method efficiently selects subsets of diverse image slices for expert annotations in a way that the most information-contributing slices are ranked most highly, allowing to train image segmentation models from high-sparsity ratio annotations. In the KCB-Net, three 2D segmentation modules and one 3D module integrating features across multiple scales with edge-aware branches are ensembled to generate pseudo-labels of the un-annotated slices, which are then used to re-train the 3D model. An IPM process is employed to post-process the probability maps generated by the 3D model. Experiments on two large knee datasets show that our new approach outperforms state-of-the-art methods on fully annotated data, and can notably improve segmentation performance when

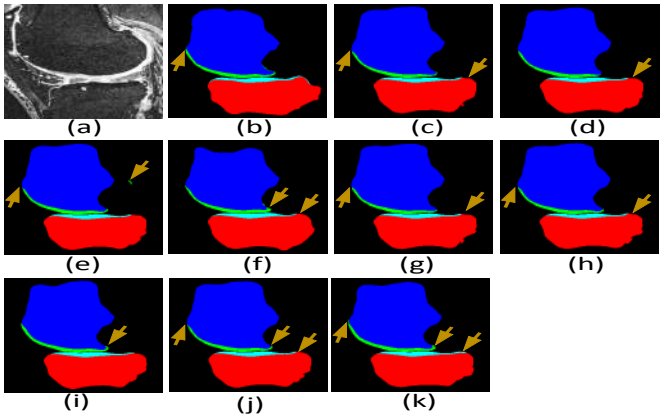


Fig. 13: Visual comparison of component-specific contributions (S1–S8) in our method in the sagittal view on the Iowa dataset. (a) An input 2D slice from a 3D image; (b) ground truth; (c) segmentation obtained by the ensemble method [10]; (d)-(k) segmentations obtained using the S1–S8 components, respectively. Note that our method successfully segments even thin cartilage areas. Arrows point to some spots of interest.

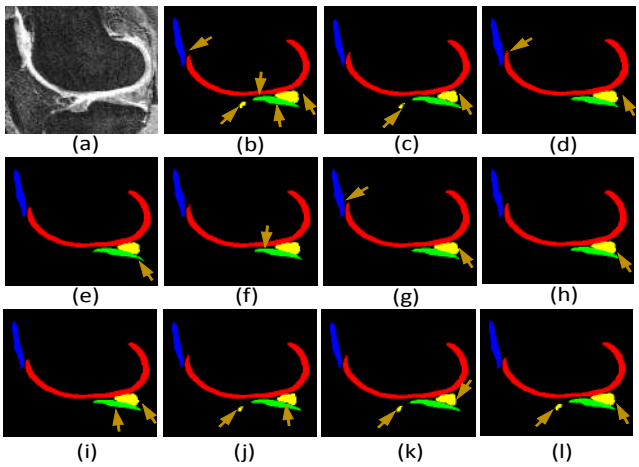


Fig. 14: Visual comparison of component-specific contributions (S1–S8) in our method with the UAD and ensemble [10] methods in the sagittal view on the iMorphics dataset. (a) An input 2D slice from a 3D image; (b) ground truth; (c) segmentation obtained by the ensemble method [10]; (d) segmentation obtained by UAD; (e)-(l) segmentations obtained using our S1–S8 components, respectively. Note that our method can segment the meniscus. Arrows point to some spots of interest.

annotating only small data subsets.

REFERENCES

[1] Paley Orthopedic & Spine Institute, "Anatomy of the knee joint," https://paleyinstitute.org/centers-of-excellence/cartilage-repair/anatomy-of-the-knee-joint/, 2018.

[2] Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka, "LOGISMOS—layered optimal graph image segmentation of multiple objects and surfaces: Cartilage segmentation in the knee joint," IEEE Transactions on Medical Imaging, vol. 29, no. 12, pp. 2023–2037, 2010.

[3] S. Kashyap, H. Zhang, K. Rao, and M. Sonka, "Learning-based cost functions for 3-D and 4-D multi-surface multi-object segmentation of knee MRI: Data from the osteoarthritis initiative," IEEE Transactions on Medical Imaging, vol. 37, no. 5, pp. 1103–1113, 2017.

[4] H. Xie, Z. Pan, L. Zhou, F. A. Zaman, D. Chen, J. B. Jonas, Y. Wang, and X. Wu, "Globally optimal segmentation of mutually interacting surfaces using deep learning," arXiv preprint arXiv:2007.01259, 2020.

[5] F. Liu, Z. Zhou, H. Jang, A. Samsonov, G. Zhao, and R. Kijowski, "Deep convolutional neural network and 3D deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging," Magnetic Resonance in Medicine, vol. 79, no. 4, pp. 2379–2391, 2018.

[6] F. Ambellan, A. Tack, M. Ehlke, and S. Zachow, "Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative," Medical Image Analysis, vol. 52, pp. 109–118, 2019.

[7] C. Tan, Z. Yan, S. Zhang, K. Li, and D. N. Metaxas, "Collaborative multi-agent learning for MR knee articular cartilage segmentation," in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2019, pp. 282–290.

[8] H. Zheng, Y. Zhang, L. Yang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen, "A new ensemble learning framework for 3D biomedical image segmentation," in AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 5909–5916.

[9] L. Yu, J.-Z. Cheng, Q. Dou, X. Yang, H. Chen, J. Qin, and P.-A. Heng, "Automatic 3D cardiovascular MR segmentation with densely-connected volumetric ConvNets," in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2017, pp. 287–295.

[10] H. Zheng, Y. Zhang, L. Yang, C. Wang, and D. Z. Chen, "An annotation sparsification strategy for 3D medical image segmentation via representative selection and self-training," in AAAI Conference on Artificial Intelligence, vol. 34, no. 4, 2020, pp. 6925–6932.

[11] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in International Conference on Medical Image Computing and Computer-assisted Intervention, 2016, pp. 424–432.

[12] H. Zheng, S. M. M. Perrine, M. K. Pitirri, K. Kawasaki, C. Wang, J. T. Richtsmeier, and D. Z. Chen, "Cartilage segmentation in high-resolution 3D micro-CT images via uncertainty-guided self-training with very sparse annotation," in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2020, pp. 802–812.

[13] J. Chen, L. Yang, Y. Zhang, M. Alber, and D. Z. Chen, "Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation," Conference on Neural Information Processing Systems, pp. 3036–3044, 2016.

[14] Z. Guo, H. Zhang, Z. Chen, E. van der Plas, L. Gutmann, D. Thedens, P. Nopoulos, and M. Sonka, "Fully automated 3D segmentation of MR-imaged calf muscle compartments: Neighborhood relationship enhanced fully convolutional network," Computerized Medical Imaging and Graphics, vol. 87, p. 101835, 2021.

[15] P. Liang, Y. Zhang, Y. Ding, J. Chen, C. S. Madukoma, T. Weninger, J. D. Shrout, and D. Z. Chen, "H-EMD: A hierarchical earth mover's distance method for instance segmentation," submitted, 2021.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in IEEE Conference on Computer Vision and Pattern recognition, 2016, pp. 770–778.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[18] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2017, pp. 399–407.

[19] L. Zhou, Z. Zhong, A. Shah, B. Qiu, J. Buatti, and X. Wu, "Deep neural networks for surface segmentation meet conditional random fields," arXiv e-prints, pp. arXiv–1906, 2019.

[20] S. Sun, M. Sonka, and R. R. Beichel, "Graph-based IVUS segmentation with efficient computer-aided refinement," IEEE Transactions on Medical Imaging, vol. 32, no. 8, pp. 1536–1549, 2013.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., "PyTorch: An imperative style, high-performance deep learning library," arXiv preprint arXiv:1912.01703, 2019.

[22] E. Panfilov, A. Tiulpin, S. Klein, M. T. Nieminen, and S. Saarakkala, "Improving robustness of deep learning based knee MRI segmentation: Mixup and adversarial domain adaptation," in IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 450–459.

2D to 3D Evolutionary Deep Convolutional Neural Networks for Medical Image Segmentation

Tahereh Hassanzadeh, *Student Member, IEEE*, Daryl Essam, *Member, IEEE*, and Ruhul Sarker, *Member, IEEE*

***Abstract*— Developing a Deep Convolutional Neural Network (DCNN) is a challenging task that involves deep learning with significant effort required to configure the network topology. The design of a 3D DCNN not only requires a good complicated structure but also a considerable number of appropriate parameters to run effectively. Evolutionary computation is an effective approach that can find an optimum network structure and/or its parameters automatically. Note that the Neuroevolution approach is computationally costly, even for developing 2D networks. As it is expected that it will require even more massive computation to develop 3D Neuroevolutionary networks, this research topic has not been investigated until now. In this article, in addition to developing 3D networks, we investigate the possibility of using 2D images and 2D Neuroevolutionary networks to develop 3D networks for 3D volume segmentation. In doing so, we propose to first establish new evolutionary 2D deep networks for medical image segmentation and then convert the 2D networks to 3D networks in order to obtain optimal evolutionary 3D deep convolutional neural networks. The proposed approach results in a massive saving in computational and processing time to develop 3D networks, while achieved high accuracy for 3D medical image segmentation of nine various datasets.**

***Index Terms*— 2D medical image segmentation, 3D medical image segmentation, deep convolutional neural network, evolutionary computation, neuroevolution.**

I. Introduction

A CONVOLUTIONAL Neural Network (CNN) [1] is a type of Neural Network (NN) that has basically been developed for image processing, and applied to various applications, such as image classification [2] and segmentation [3]. A CNN is constructed based on mimicking various layers of operations for feature extraction and prediction. With regards to the stated application, the network topology and its corresponding parameters need to be specified. Designing a Deep Convolutional Neural Network (DCNN) is a challenging task because with increasing the depth of the network, the number of parameters that need to be set up will also be increased. As a result, the arrangement of the layers and setting the parameters are tedious tasks. Besides, DCNNs, in most cases, using a massive number of trainable parameters consequently increases the required computation and the probability of overfitting [4].

The methods for 2D image processing are widely available. However, due to the advancement of medical image technologies, there is a demand of 3D images for analysing the status of critical organs. Designing a 3D DCNN is much more complicated than the same for 2D because it requires a significantly higher number of parameters. A network with more parameters needs more data for training. However, most of the available 3D datasets have a limited number of 3D volumes. These issues limit the development of 3D deep neural networks and the possibility of applying automatic methods to design a network's structure more precisely.

Neuroevolution is an optimisation technique, which is based on evolutionary computation, that is used to decide an appropriate neural network topology and/or its parameters. Examples of various applications can be found in [5]–[10]. In Neuroevolution, because of evolutionary algorithms' ability to automatically design a neural network, it has become a very popular approach to do this. However, Neuroevolution is computationally very expensive, specifically when developing an optimum evolutionary 3D network, so these high computational requirements may limit the use of such an approach in many practical cases.

A study by IDC (International Data Corporation), named Age Data 2025 [11], estimates that the globally produced data will grow to 165 zettabytes by 2025 which is ten times the data that was generated in 2016. Another study by Stanford Medicine [12] published a white paper that forecasted that the amount of healthcare data would be 2314 exabytes by 2020, while the worldwide healthcare data was just 153 exabytes in 2013. These studies show the tremendous worldwide data growth, specifically in healthcare. Besides, according to International Business Machines (IBM) researcher's estimation, medical images are at least 90 per cent of all medical data [13], making them the most prominent data in healthcare enterprises. Dealing with the huge amount of medical images

Manuscript received August 27, 2020; revised October 1, 2020; accepted October 28, 2020. Date of publication November 3, 2020; date of current version February 2, 2021. This work was supported by the Australian Government, undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI). *(Corresponding author: Tahereh Hassanzadeh.)*

The authors are with the Canberra Evolutionary Optimization Research Group, University of New South Wales, Canberra, ACT 2612, Australia (e-mail: t.hassanzadehkoohi@student.unsw.edu.au; d.essam@adfa.edu.au; r.sarker@adfa.edu.au).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2020.3035555

becomes overwhelming for radiologists, especially in some hospitals where they are faced with thousands of images daily. Therefore, there is a need for automatic methods to extract information from medical images. One of the most widely used methods for medical image analysis is image segmentation to find a specific organ or abnormality in an image. Since multidimensional (3D, 4D etc.) images are the most prominent type of medical images, and DCNNs are one of the best image processing methods, developing optimum and accurate networks to deal with medical images seems necessary.

In this article, to deal with 3D medical image segmentation, we propose two new approaches as follows. The first one is an evolutionary 3D network and the second one is to use an appropriate 2D network for 3D medical image segmentation. Therefore, a new evolutionary approach is introduced for 2D medical image segmentation, and then the 2D models are converted to 3D models for 3D volume segmentation. As indicated earlier, we have also developed an evolutionary 3D model to create 3D networks and compared the results with converted 3D networks. To the best of our knowledge, there is no 3D Neuroevolutionary approach to generate a 3D network using a population of 3D networks until now. Also, to the best of our knowledge, no attempt has been made to use 2D images and an evolutionary 2D strategy to develop a 3D network for 3D image processing. It is appropriate to mention here that we apply a Genetic Algorithm (GA) that is an evolutionary computation technique to evolve networks.

The importance of this research can be described as follows. Firstly, in our proposed model, we show that 2D image slices are representative enough to be utilised to develop a 3D model without using 3D volumes, as our experimental results show that a network structure that is created using 2D image slices could process the corresponding 3D volumes with high accuracy. Secondly, our proposed network shows the capability of evolutionary computation to develop a new deep network even without using the original data format. Finally, our proposed approach has addressed the problem of the limited computational facilities available for developing 3D networks by showing a huge saving of computational time. The analysis of results demonstrates high accuracy for segmentation of nine various publicly available medical segmentation datasets, and also a massive saving in the required computation and processing time.

The rest of the paper is organised as follows. The next section provides an overview of Neuroevolution. Section 3 demonstrates the proposed model. The dataset and experimental results are discussed in section 4. Finally, section 5 provides the discussion and conclusion.

II. Overview of Automated Network Search

Manually designing a neural network needs deep knowledge about the neural network as well as large computation facilities and time, as it requires many trials to obtain the best topology for a given dataset. One of the solutions to this is to design a network automatically using evolutionary computation. Utilising an evolutionary neural network as an automatic method to develop a network was introduced by Montana and Davis [14], with utilising a Genetic Algorithm (GA) [15], for initialisation of a network's weights. Later work has also focused on finding network's topology [16].

The successful applications of Neuroevolutionary in developing feed–forward networks has convinced researchers to apply evolutionary computation for finding Deep Neural Networks' (DNNs) and Deep Convolutional Neural Networks' (DCNNs) network topologies and/or their corresponding parameters. However, developing evolutionary CNN or DNN also needs significant computational facilities to find the specified parameters, but using an appropriate evolutionary technique with an adaptation ability can somewhat address the problem.

Koutník *et al.* [17] applied evolutionary computation to a deep network for the first time in 2014. Where they applied Neuroevolution to compress a neural network for vision–based Reinforcement Learning (RL). From 2014 to the present, several papers have been published about using evolutionary computation for a network's evolution. For example, Miikkulainen *et al.* [18] proposed CoDeepNEAT as an automatic method for the evolution of a deep neural network along with its parameters using NeuroEvolution of Augmenting Topologies (NEAT) [19]. NEAT is a graph–based technique, where evolution starts with a population of small networks with minimal complexity, and then during the evolution, more complex networks are created applying mutation and crossover. The differences of CoDeepNEAT to NEAT are the way they use nodes and edges. In CoDeepNEAT each node represents a layer as well as a table that contains its corresponding parameters, but in NEAT, each node shows a neuron. Also, the edges represent how nodes should be connected together; however, in NEAT, each edge corresponds to a weight. CoDeepNEAT is constructed from two populations of modules and blueprints that should be evolved separately. Whereas, each module is a block with a relatively complex structure, and a blueprint is a graph that shows how modules should be connected. CoDeepNEAT is applied in different applications such as image classification, image captioning, and object recognition and has obtained competitive results to hand–designed networks. However, it needs a couple of days and hundreds of GPUs for evolution.

EvoDeep is another graph–based evolutionary approach for automatically developing a deep network structure along with its parameters. It was introduced by Martín *et al.* [20]. EvoDeep is constructed from a population of graphs that are created using a Finite State Machine (FSM). An FSM has a start and an end node to specify the first and last layer of a network and the other nodes determine the rest of the layers. The edges pinpoint the possible transactions between layers, including dense, dropout [4], reshape, convolution, flatten, and max–pooling layers. EvoDeep achieved high accuracy in an image classification application.

EvoCNN is an evolutionary DCNN for an image classification application that was developed by Sun *et al.* [21]. EvoCNN uses a variable–length encoding to create chromosomes and consequently networks with various lengths. Each chromosome is encoded using three different types of genes, namely convolution layer, pooling layer, and full connection layer. Furthermore, each gene is also encoded with several

parameters. For instance, the convolution layer's embedded information includes filter size, stride width and length, and the number of filters. In EvoCNN, a Genetic Algorithm (GA) equipped with a new crossover strategy is utilised for a network's evolution.

Besides, evolutionary synthesis of DNN as a nature–inspired methodology was proposed by Shafiee *et al.* [22] for image segmentation. Their proposed model mimics the three biological evolution operations of heredity, natural selection, and random mutation. The syntactic evolution starts with a network using a specified number of nodes and full connections such that there are edges between each node in the current layer to all other nodes in the next layer. In each generation, a new offspring network is synthesised stochastically so that the number of nodes stayed unchanged, but the number of edges is reduced. The aim is to find a minimal network with high accuracy. According to the authors, this strategy can be applied to existing network architectures as an optimisation strategy. The synthesis strategy was applied to VGG16 [23] and achieved high accuracy for image segmentation.

Genotype encoding is a critical issue that can affect a network's evolution. Some papers introduced new encoding strategies as new evolutionary techniques for network's evolution. For instance, Baldominos *et al.* [24] proposes a new evolutionary technique with two different encoding approaches for handwriting recognition. In the first approach, a binary representation is utilised for the encoding of genotypes, and consequently, a Genetic Algorithm (GA) is used to evolve the networks. In the second approach, Grammatical Evolution (GE) [25] is applied to the network's evolution. GE provides a more flexible representation of phenotypes and also an infinite language to generate networks. Their experimental results show that using GE outperformed GA and thus showed the importance of using an appropriate encoding strategy.

Another encoding strategy that was inspired by Internet Protocol (IP) encoding was also proposed for genotype encoding [26]. The IP–based Particle Swarm Optimisation (PSO) [27] method was developed for image classification such that each network is encoded using an IP like binary pattern. Since each network for an image classification application is constructed from convolution, pooling, and fully connected layers, and each layer has its features along with a range of corresponding values, a number of bits have been calculated for each feature. For instance, the range of one to eight filters is considered for each convolution layer; as a result, three bits are needed to show filter size. In their proposed model, collectively 45 bits are used to construct a network's genotype. Also, to specify the type of layer in the IP representation, subnet patterns are utilised. PSO is compatible with this representation to evolve networks because it is a kind of fixed–length encoding that can develop various depth networks. Also, a variable–length IP encoding version was introduced by Wang *et al.* [28]. In the modified IP representation version, the maximum length of the network is not set up in advance, and also a new cross–over strategy is developed to deal with various length genotypes.

Reinforcement Learning (RL) [29] and Recurrent Neural Networks (RNNs) [30] can also be used to develop a new network's structure automatically. Neural Architecture Search (NAS) [31] is an RNN–based learning model that can create a network for image classification. In the proposed model, a RNN is utilised to generate a model description for the neural network, and then the RL is applied to improve the model. Such that a controller which is implemented as an RNN is utilised to generate hyper–parameters to create a network. Model's based on NAS search have also been developed [32]. For example, NASNet architecture was developed based on the NAS strategy by Zoph *et al.* [33]. NASNet creates a network using two types of cells (blocks), namely normal cells and reduction cells. The architecture of the network is constructed from a stack of these two types of cells. The network architecture is fixed; however, the structure of the cells aims to learn to effectively utilise a RNN. The NASNet strategy decreases the required computation by changing the search space. In NASNet, the evolution process is done in a smaller dataset to find the cells, and then stacks of the obtained cells are utilised to construct the final network structure for classification of the larger dataset. Furthermore, AmoebaNet [34] is an evolutionary strategy that was introduced based on NAS for image classification. AmoebaNet also followed the NAS strategy to develop a network structure using normal and reduction cells, however, an evolutionary strategy was used to develop cells instead of RNN. Also, following the NASNet method, the cells are created using the smaller dataset. Besides, they proposed a novel strategy for tournament selection, such that younger genotypes should be selected instead of the best one, which improved its classification accuracy.

NAS strategy is also utilised to improve networks for image segmentation applications. Dong *et al.* [35], introduced a NAS–based model to create an Adversarial neural network for medical image segmentation, such that the NAS technique is applied to create an optimal discriminator. As, a fully connected network is used for segmentation and the NAS strategy is applied to find an optimum cell to create a discriminator. Another NAS–Based model for image segmentation was introduced by Weng *et al.* [36] and named NAS U–Net. In their proposed model, a fixed U–Net based network (four blocks in each side of the network) is considered such that the blocks structures are found using the NAS model. To find the optimal down cell and up cell, the PASCAL VOC 2012 dataset [37] is utilised. However, for final evaluation, the found network is used for medical image segmentation. Besides, Mortazi and Bagci [38] proposed an RL–based model to generate an automatic network for medical image segmentation. The Policy Gradient (PG) [39] algorithm, which is a kind of RL model is applied to learned hyper–parameters of the network. So that, each parameter considered a policy that needs to be found during training.

As discussed above, most of the automatic frameworks need to be parallelised over hundreds of worker machines for evolution. LEAF (Learning Evolutionary AI Framework) [40], is an evolutionary auto–machine learning framework that can optimise a network's structure, its hyper–parameters, and its size, using evolutionary frameworks such as CoDeepNEAT [18]. LEAF is a framework that enables parallel processing and

is constructed from three layers: problem domain, algorithm, and system layer. The problem domain layer is responsible for solving three problems, namely hyper–parameter, architecture, and the network's size. LEAF's algorithm layer considers DNN parameters and structure evolution. Besides, the system layer can parallelise the training of DNNs on multiple machines. LEAF using CoDeepNEAT obtained high accuracy for medical image classification and natural language analysis. Except for using evolutionary computation and reinforcement learning, there are other models to generate a network automatically, such as using a differentiable approach [41] and continued optimisation [42]. Both tried to propose more efficient strategies to find networks for image classification applications.

In this section, we have reviewed some of the most critical and state–of-the-art techniques to develop automatic deep neural networks. As showed above, all the automatic models used various encoding and evolutionary and RNN techniques to develop 2D networks for 2D image processing.

A. Research Contributions

- For the first time, an evolutionary 3D model has been developed for medical image segmentation.
- An evolutionary approach has been proposed to develop 2D networks for 2D medical image segmentation.
- Introduced a 2D network based 3D evolutionary network for medical image segmentation. To the best of our knowledge, no such attempt has been made in any previous image processing application.
- For the first time, we analysed whether a network topology that was developed using 2D images can segment 3D volumes with acceptable accuracy.
- We are able to achieve high accuracy for 3D medical image segmentation using an evolutionary 2D network structure.
- The proposed approach decreases the required computation and processing time significantly for developing 3D networks.

III. Proposed Model

The aim of this article is to investigate the possibility of creating 3D networks from evolutionary 2D DCNNs. Firstly, we propose an evolutionary approach to find an optimum topology of 2D networks. Then, we propose a technique to convert evolutionary 2D networks to 3D networks. We also develop evolutionary 3D networks based on the proposed evolutionary approach.

In this article, a U–Net–based [43] evolutionary strategy is proposed to create evolutionary U–Net–based networks, which uses a fixed–length block–based genotype encoding model to create variable–depth networks. To create a 2D network architecture, twelve parameters need to be optimised using Genetic Algorithm (GA), namely: number of blocks, number of convolution layers in each block, number of filters, filter size, dropout [4], pooling, activation function, shortcut connections, long connections [44], Batch Normalisation (BN) [45], optimiser, and learning rate. Our proposed evolutionary method is a block–based method, which evolves the network block by block based on the hyper–parameter values.

TABLE I
The Hyper–Parameters and Their Corresponding Ranges That Are Used to Create a Network

Hyper–parameters	Range
Number of blocks	5
Number of convolution layers per block	$[1-3]$
Number of filters	$[8, 16, 32, 64]$
Filter size	$[3\times3, 5\times5, 7\times7]$
Dropout	$[0-0.7]$
Pooling	[Averagepooling (0), Maxpooling (1)]
Activation function	[Sigmoid (0), Relu (1)]
Shortcut connection	$[0, 1]$
Long connection	$[0, 1]$
Batch Normalisation	$[0, 1]$
Optimiser	[adam, rmsprop, adagrad, adadelta]
Learning rate	$[0.1, 0.01, 0.001]$

Table I provides parameters along with their corresponding ranges that are used to create a network block by block. In our proposed genotype encoding model, each chromosome is shown with five blocks, so each chromosome has 52 genes, where 50 of them are used to create five blocks and two of them to find the appropriate optimiser and a learning rate for training the network. The number of blocks indicates the number of blocks in the down–sampling part of the network along with a bridging block. The maximum number of blocks is set to five because the 2D networks will be used for evaluation of 3D volumes which are re–sliced to a minimum of 16 to 64 slices for different datasets. Therefore, five blocks are the maximum number of blocks that can be used in the down–sampling and bridging section collectively. Then the up–sampling blocks will be created using the down–sampling block's parameters. In the following, we use an example to describe how a Genetic Algorithm (GA) is utilised to perform the network's evolution.

Since our proposed model is blocked–based, we need to construct a network block by block. Therefore, ten parameters are used to represent each block. All the parameters need to be initialised randomly and then be updated based on GA operations. An example of a block's genotype (see Figure 1a) and its corresponding Phenotype (see Figure 1b) are demonstrated in Figure 1. As can be seen from Figure 1a, the first gene shows whether the block is active (1) or not (0), all active blocks will be included into the structure of the network. The block's genotype shows that the block is active and there is a shortcut connection to copy the block's input feature maps to its output; also a long connection is between this block in the down–sampling part and its corresponding block in the up–sampling section. There is one 2D convolution layer with a filter size of 7×7, and with 32 feature maps as the output of the block. The activation function is ReLU, and Batch Normalisation [45] should be applied to the output features. The dropout [4] rate and max–pooling are also specified for the block. To stack blocks on top of each other the long connection parameter in the bridging block and up–sampling blocks should be ignored. It needs to be noted that in the proposed model convolution and pooling stride are kept fixed as two.

After converting each genotype to its corresponding network and training the networks, the obtained accuracy shows the fitness value of each chromosome. To create a new generation, the best chromosome moves to the next generation directly,

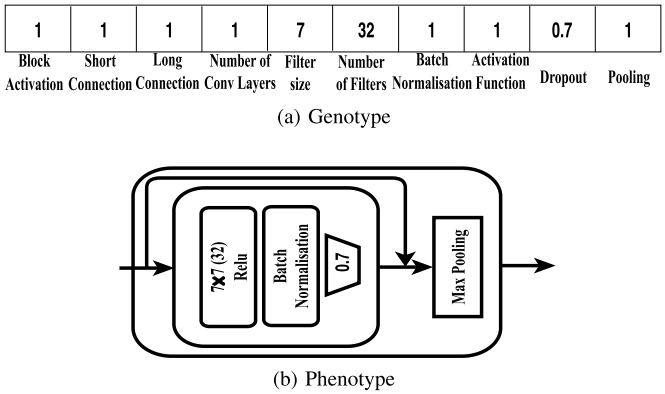


Fig. 1. An example of the Genotype and Phenotype of a block.

and the rest of the population are selected using the Roulette wheel selection method [46] and the new generation is created using a single–point crossover operation. Next, mutation is applied to the chromosomes to increase the exploration ability of the GA. For each chromosome, a random number ranged from zero to three specifies the number of mutations in each chromosome, which is a random change in randomly selected genes in regards to its stated range. In the end, a swarm of superior 2D networks will be selected to convert 2D networks to 3D networks. This process is also repeated to create a population of 3D networks using 3D swarms – detailed analysis is provided in the next section.

IV. Experiments

A. Dataset and Pre–Processing

For the evaluation of the proposed model, in the first stage PROMISE12 [47], a publicly available prostate MRI segmentation dataset is used. In PROMISE12, 50 MRI volumes (1377 image slices) were collected from four different medical centres. The MRI volumes have various qualities, to improve their qualities the Z–score method (zero mean unit variance) [48], [49] is applied. Since, in the first stage we need to develop 2D networks, the 2D slices are used to develop evolutionary neural networks. Among the 50 MRI volumes, all image slices corresponding to 40 volumes were used for training, five of them for validation and the rest of them for testing. In regards to the limited available training images, various augmentation methods, including horizontal flip, vertical flip, rotation (randomly rotates the image clockwise from 0 to 10 degrees), zoom (uniformly randomly sample from 1 to 1.2), and elastic transformation [50] were applied to increase the number of training images to 50000 to improve the quality of the network's evolution. For the evolutionary 3D model, the volumes were re–sliced to $16 \times 128 \times 128$, and then augmented to 2000 volumes.

B. Implementation

The Keras python package [51] was utilised for implementation of the proposed model. All experiments were carried out on four Nvidia GPUs along with CPUs. Table II provides the parameters that were used for the 2D network's evolution. As can be seen from Table II (2D column), to increase the diversity of the population, the algorithm is initialised with 60 genotypes, and in the next generations, the size of the population is reduced to 30 and training continued to nine generations. Each 2D network is trained to five epochs during the evolution. It needs to be emphasised that the population size, number of generations, batch size, and augmentation size were based on preliminary experiments. Also, all the experiments were repeated three times. Besides, Table II (3D Column) provides the range of the parameters for the evolutionary 3D model, where the 3D volumes, instead of 2D image slices, was utilised for the 3D network 's evolution. The selection criteria during the evolution and also to find the best networks at the end of training, is the segmentation accuracy of the validation set. As can be seen from Table II, batch size, initial population size, population size, and the number of augmentations are reduced in comparison to the 2D model, because of the expensive computation costs of the 3D models. Also, for the 3D models, the filters for convolution, pooling, and deconvolution were converted from 2D to 3D. For example, the 3×3 filter was converted to $3 \times 3 \times 3$.

TABLE II
The General Parameters and Their Corresponding Values for Training the Evolutionary 2D and 3D Models

Training Parameters	Ranges	
	2D	3D
Number of generations	9	9
Number of epochs	5	5
Number of runs	3	3
Early stopping	3	3
Batch size	16	4
Initial population size	60	12
Population size	30	12
Number of Augmentation	50000	2000

The results in the training stage have been evaluated using the Dice Similarity Coefficient (DSC) [52] that can be seen in equation 1). Where, Y' shows the label image, Y represents the predicted segmented image, and $|Y'|$ and $|Y|$ indicate the cardinality of Y' and Y. Moreover, in the proposed model, DSC was used as the loss function for training the networks.

$$DSC = \frac{2 \mid Y' \cap Y \mid}{\mid Y' \mid + \mid Y \mid} \tag{1}$$

Finally to compare the proposed model to previous works, Hausdorff distance [53], Average Boundary Distance (ABD) [53], and absolute Relative Volume Difference (aRVD) [54] are also employed.

C. Experimental Results

1) 2D Networks: After training the proposed model to nine generations using the above setup, the ten best 2D networks that obtained the highest validation accuracy were selected from the best run. All these ten networks were then trained for five epochs using 50000 image slices and a batch size of 16 during the evolution stage. The DSCs of the ten best networks after five epochs using the test set are presented in Table III (third column). As can be seen from Table III (third column), the best 2D network obtained 0.81 DSC for segmentation of the test images.

2) Converted 3D Networks: As discussed above, the aim of our proposed model is analysing the possibility of using evolutionary 2D networks to develop 3D networks. Therefore, we converted the ten–best 2D networks to 3D networks, where the number and the type of layers and other parameters were

TABLE III
COMPARISON OF 2D NETWORKS AND THEIR CORRESPONDING CONVERTED 3D NETWORKS IN TERMS OF OBTAINED DSCS AND THE NUMBER OF TRAINABLE PARAMETERS (NTP)

Networks	2D Networks			Converted 3D Networks	
	NTP	Mean DSC-All slices	Mean DSC-Selected Slices	NTP	Mean DSC
Network 1	2, 112, 849	0.75	0.70	12, 758, 417	**0.82**
Network 2	1, 479, 873	0.75	0.74	9, 873, 153	**0.89**
Network 3	1, 639, 297	0.74	0.64	9, 180, 609	**0.85**
Network 4	213, 157	0.72	0.62	734, 666	**0.83**
Network 5	1, 612, 993	0.74	0.54	10, 733, 825	**0.79**
Network 6	2, 491, 057	0.80	0.72	14, 956, 337	**0.81**
Network 7	2, 870, 017	0.81	0.72	16, 170, 881	**0.86**
Network 8	2, 870, 017	0.59	0.45	16, 170, 881	**0.83**
Network 9	1, 812, 369	0.75	0.73	8, 503, 313	**0.87**
Network 10	3, 140, 321	0.65	0.74	17, 596, 257	**0.82**

kept unchanged. What we did was just converting 2D operations to 3D operations, including convolution, deconvolution, and pooling layers. With converting the 2D networks to 3D networks the number of trainable parameters increased five to six times in each network (see Table III). For training the 3D networks, we re–sliced and re–sized all 3D volumes to $16 \times 128 \times 128$; this means each volume has 16 image slices. The 16 middle slices of each volume were selected during a re–slicing process. Besides, because of the limited number of volumes, augmentation was applied, and the number of volumes was increased to 32000. All ten converted 3D networks were trained for just five epochs using a batch size of 16, and the obtained results are demonstrated in Table III (last column).

As mentioned above, for the training of the 3D models, we did not use all the image slices. For a fair comparison, we again trained the best 2D networks using the image slices used for training the 3D networks and also with the same number of augmentations and batch size. The results are also in Table III (fourth column). As can be seen from Table III (fourth column), with decreasing the number of slices, the accuracy of the 2D networks reduced. But the results of the 3D networks are remarkably more accurate than the corresponding 2D networks, such that the best converted 3D network gained 0.89 DSC for segmentation of the 3D volumes. Besides, the worst network obtained 0.79 DSC. Overall, the converted 3D networks work well for 3D volumes segmentation.

3) Evolutionary 3D Networks: In this section, to compare the evolutionary 3D networks to converted 3D networks, we also examined the 3D network's evolution using 3D volumes. Since evolving 3D networks using 3D volumes is very time–consuming and computationally demanding, for 3D network's evolution we decreased the number of augmentations to 2000, batch size to four, and population size to 12 (see Table II). However, all other parameters were kept the same as above, and training continued to nine generations – similar to the evolutionary 2D strategy. In the end, we selected the ten best evolved 3D networks (the networks that obtained the highest validation accuracy), and the results are presented in Table IV (second column). In Table IV, the first DSC indicates the DSCs of the ten best networks trained by 2000 volumes after five epochs, and the second DSC presents the DSCs of the same networks trained by 32000 volumes, batch size of 16, to five epochs, similar to the 3D converted networks.

As shown in Table IV, the best 3D network after nine generations obtained 0.66 DSC and the worst one almost

TABLE IV
THE DSC AND NUMBER OF TRAINABLE PARAMETERS (NTP) OF THE TEN–BEST EVOLUTIONARY 3D NETWORKS AFTER NINE GENERATIONS. THE FIRST DSC IS OBTAINED USING 2000, AND THE SECOND DSC IS ACHIEVED USING 32000 TRAINING VOLUMES

Networks	2000	32000	NTP
Network 1	0.420	0.454	2, 776, 297
Network 2	0.457	0.484	1, 988, 857
Network 3	0.417	0.404	2, 776, 297
Network 4	0.14	0.70	1, 670, 233
Network 5	0.41	0.522	2, 790, 153
Network 6	0.61	**0.79**	2, 150, 897
Network 7	**0.66**	0.663	2, 035, 609
Network 8	0.073	0.101	1, 770, 073
Network 9	0.62	0.757	2, 102, 705
Network 10	0.40	0.416	1, 365, 913

TABLE V
COMPARISON OF THE BEST EVOLUTIONARY 2D AND 3D NETWORKS VERSUS CONVERTED 3D NETWORK FOR PROSTATE MRI SEGMENTATION

Models	DSC	HD	ABD	aRVD
Best evolutionary 2D network	0.81	25.77	16.87	17.01
Best evolutionary 3D network	0.79	28.83	18.27	19.66
Best converted 3D network	**0.89**	12.34	7.56	3.08

zero for the MRI volume segmentation. The structure of Network 8 that obtained the worst results seems reasonable; in that the sequence of layers is correct; however, the network training failed and could not achieve high accuracy for the segmentation of the validation and test sets. Note that we re-trained the network to assure that the failure was not because of random initialisation; however, the results stayed almost unchanged. It is evident that the obtained results are not good enough after nine generations. The reasons for this may be the limited population size and the augmentation size that were used for the network's evolution, which were because of the substantially increased computation required for the 3D evolution. Besides, even with increasing the number of training volumes to 32000, the segmentation accuracy is still low, and the best evolutionary 3D network achieved 0.79 DSC, that is still 10% lower than the best converted 3D network.

Furthermore, the number of trainable parameters of the ten–best evolutionary 3D networks are included in Table IV. As can be seen, they are small networks in terms of the number of trainable parameters; however, they are not generalised enough to segment unseen images.

4) Comparison of the Best Evolutionary 2D and 3D Networks Versus Converted 3D Network: In this section, the best evolutionary 2D and 3D networks are compared with the converted 3D network using DSC, HD, ABD, and aRVD criteria (see Table V). As shown, the best converted 3D network is more accurate than the best evolutionary 2D and 3D networks. As can be seen from the results, the best 2D evolutionary network outperforms the best 3D network. The results show that developing 3D networks using evolutionary 2D networks is a secure method for 3D network's development and consequently 3D segmentation. Furthermore, the results show that 2D image slices are representative and informative enough to be used to develop 3D networks.

We also provide some examples as a subjective comparison of the best networks. As shown in Figure 2, three volumes are illustrated such that the red volume is the ground truth, and the cyan one is the predicted volume. The first row represents

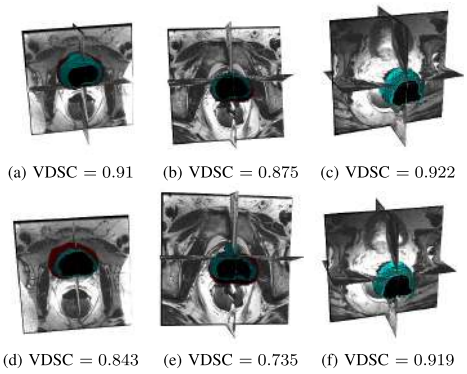


Fig. 2. Three sample segmented volumes. The red volume is the ground truth, the cyan volume is predicted volume using the best converted 3D network (first row), and the best evolutionary 3D network (second row). Volumetric DSC(VDSC).

the obtained results of the best converted 3D network, and the second row shows the segmentation results of the best evolutionary 3D network using the same images. The volumetric DSC (VDSC) corresponding to each volume shows the accuracy of segmentation. It can be seen that the best converted 3D network (first row) segments the prostate more accurately.

D. Extra Experiments

In this section to show the capability of our proposed model, we examined our proposed model for the segmentation of eight more datasets. The detailed analysis is provided below.

1) Datasets: Besides, the PROMISE12 [47] dataset, eight other publicly available datasets are used to show the capability of our proposed model. Three CT liver segmentation datasets: CHAOS [55], SLIVER07 [53], and Decathlon–Liver [56], and also, Decathlon–Spleen (CT) [56], Decathlon–Pancreas (CT) [56], Decathlon–Prostate (MRI) [56], Decathlon–Hippocampus (MRI) [56], and Decathlon–Heart (MRI) [56] segmentation datasets are used for evaluation of the proposed model. The detailed information of all datasets can be seen in Table VI. The 2D image slices are used for training the proposed evolutionary model to find the 2D networks. Then the 3D volumes are utilised for training the converted 3D models. The 3D volumes are shown as $N \times M \times X \times Y$, where N is the number of volumes, M denotes the depth (number of slices), X is the width, and Y is the height of a volume. The number of slices in the 3D volumes have been chosen based on the number of slices of the original volumes and also the number of slices that contain ROI. First, all the images were resized to 128×128 and then as can be seen from Table VI, in some cases the whole image and in the rest, image patches which include ROI are used for training. In the proposed model, we applied volumes that include ROI for training, testing, and the validation of the models.

2) Experimental Results: The DSCs of the ten best networks in 2D and 3D for eight segmentation datasets are provided in Table VII. Based on our preliminary experiments using the PROMISE12 dataset, we trained the proposed model using the eight other datasets to find the best ten 2D networks for segmentation. It means we repeat experiments for each dataset separately to find the best networks for each dataset. Then

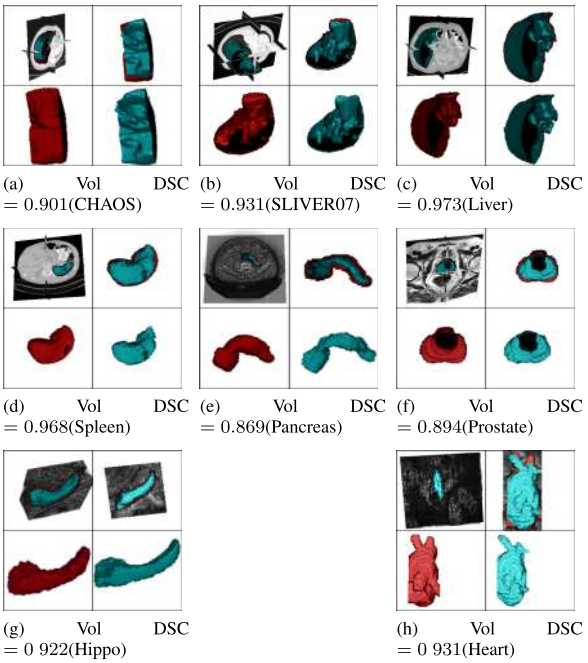


Fig. 3. Eight sample segmented volumes in regards to the eight datasets. The red volume is the ground truth, the cyan volume is predicted volume using the best converted 3D networks.

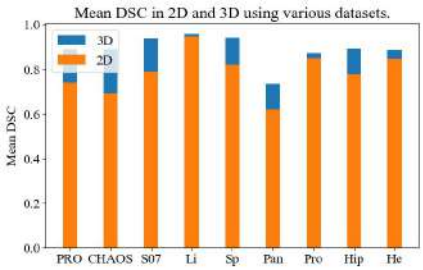


Fig. 4. The DSC of the best 3D networks and the corresponding 2D networks for nine different datasets.

we converted the best 2D networks to their corresponding 3D networks and trained the networks using 3D volumes. All networks, either in 2D or 3D, were trained for five epochs. As can be seen from Table VII, in all eight cases the best converted 3D networks outperformed their corresponding 2D networks for segmentation. In most of the cases, such as Spleen and Pancreas segmentation, there is a considerable difference between the 2D and 3D network's accuracy. However, for Liver segmentation, there is only a small difference. Also, we provide segmented sample images as subjective evaluation. As can be seen from Figure 3, a sample segmented image corresponding to each dataset is provided, where the red volume shows the ground truth, and the cyan volume shows the output of the best converted 3D network. The output segmentation results again confirm the accuracy of the proposed framework for 3D image segmentation.

The accuracy of the best converted 3D networks and their corresponding 2D networks are demonstrated in Figure 4. The orange bar indicates the accuracy of the 2D network and the blue section shows the accuracy of its corresponding converted 3D network. As shown in Figure 4, there is a

TABLE VI
THE LIST OF DATASETS AND THE NUMBER OF IMAGES IN 2D AND 3D IN TRAIN, TEST, AND VALIDATION SETS

Dataset	Volumes	Train		Test		Validation	
		2D (Slices)	3D (Volumes)	2D (Slices)	3D (Volumes)	2D (Slices)	3D (Volumes)
PROMISE12	**50 MRI**	1115	$40 \times 16 \times 128 \times 128$	165	$5 \times 16 \times 128 \times 128$	97	$5 \times 16 \times 128 \times 128$
CHAOS	**20 CT**	2025	$44 \times 32 \times 64 \times 64$	306	$13 \times 32 \times 64 \times 64$	149	$13 \times 32 \times 64 \times 64$
SLIVER07	**20 CT**	2712	$41 \times 64 \times 64 \times 64$	538	$10 \times 64 \times 64 \times 64$	909	$13 \times 64 \times 64 \times 64$
SPLEEN	**41 CT**	2203	$52 \times 32 \times 64 \times 64$	619	$11 \times 32 \times 64 \times 64$	491	$10 \times 32 \times 64 \times 64$
PANCREAS	**281 CT**	19003	$580 \times 32 \times 64 \times 64$	3883	$131 \times 32 \times 64 \times 64$	3833	$134 \times 32 \times 64 \times 64$
PROSTATE	**32 MRI**	408	$22 \times 16 \times 128 \times 128$	94	$5 \times 16 \times 128 \times 128$	100	$5 \times 16 \times 128 \times 128$
HIPPOCAMPUS	**260 MRI**	6323	$180 \times 32 \times 64 \times 64$	1441	$40 \times 32 \times 64 \times 64$	1434	$40 \times 32 \times 64 \times 64$
LIVER	**131 CT**	13287	$323 \times 64 \times 64 \times 64$	3220	$61 \times 64 \times 64 \times 64$	2656	$71 \times 64 \times 64 \times 64$
HEART	**20 MRI**	467	$42 \times 32 \times 64 \times 64$	177	$16 \times 32 \times 64 \times 64$	183	$16 \times 32 \times 64 \times 64$

TABLE VII
THE DSCs OF THE TEN BEST NETWORKS IN 2D AND 3D IN REGARDS TO EIGHT DATASETS

Networks	CHAOS		SLIVER07		LIVER		SPLEEN		PANCREAS		PROSTATE		HIPPOCAMPUS		HEART	
	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D
Net 1	0.754	0.813	0.837	0.916	0.948	0.955	0.865	0.939	0.619	**0.734**	0.863	0.857	0.780	0.892	0.857	0.865
Net 2	0.745	0.788	0.835	0.833	0.947	**0.959**	0.860	0.926	0.610	0.612	0.861	0.854	0.778	**0.893**	0.852	0.852
Net 3	0.727	0.778	0.813	0.924	0.943	0.957	0.858	0.940	0.609	0.609	0.855	0.787	0.777	0.880	0.851	0.840
Net 4	0.703	0.823	0.805	0.874	0.943	0.948	0.822	**0.941**	0.600	0.677	0.849	0.871	0.775	0.890	0.850	0.853
Net 5	0.696	0.814	0.799	0.895	0.942	0.958	0.821	0.831	0.589	0.690	0.848	0.857	0.772	0.885	0.847	**0.887**
Net 6	0.693	**0.889**	0.788	**0.940**	0.938	0.953	0.812	0.813	0.584	0.608	0.848	**0.874**	0.767	0.888	0.847	0.846
Net 7	0.674	0.742	0.787	0.861	0.938	0.952	0.787	0.838	0.572	0.607	0.847	0.858	0.767	0.886	0.839	0.868
Net 8	0.642	0.808	0.781	0.919	0.937	0.949	0.785	0.841	0.565	0.601	0.846	0.852	0.766	0.881	0.837	0.836
Net 9	0.627	0.836	0.780	0.916	0.934	0.937	0.781	0.913	0.562	0.586	0.843	0.831	0.764	0.873	0.833	0.857
Net 10	0.546	0.816	0.767	0.916	0.932	0.949	0.769	0.846	0.562	0.576	0.833	0.823	0.763	0.879	0.830	0.843

TABLE VIII
COMPARISON OF THE PROPOSED MODEL VERSUS PRIOR WORKS USING NINE DATASETS

Models	PROMISE	CHAOS	SLIVER07	LIVER	SPLEEN	PANCREAS	PROSTATE	HIPPOCAMPUS	HEART
3D U–Net [57]	0.833	0.883	0.885	0.956	0.886	0.528	0.858	0.892	0.824
Convnet [58]	0.779	0.381	0.898	0.945	0.90	0.578	0.855	0.013	0.848
3D Dense U–Net [59]	0.721	0.328	0.826	0.938	0.778	0.465	0.817	0.199	0.843
NAS U–Net [36]	0.743	0.207	0.891	0.924	0.837	0.521	0.838	0.826	0.845
Best 2D Network	0.81	0.754	0.837	0.948	0.865	0.619	0.863	0.780	0.857
2D Network	0.74	0.693	0.788	0.947	0.822	0.619	0.848	0.778	0.847
Converted 3D Network	**0.89**	**0.889**	**0.940**	**0.959**	**0.941**	**0.734**	**0.874**	**0.893**	**0.887**

significant difference between 2D evolutionary networks and their relevant converted 3D network.

E. Comparison With Prior Works

To show the performance of the proposed model, we also compared the obtained results versus 3D U–Net [57], Convnet [58], and 3D Dense U–Net [59], that are three FCNN–based model for 3D medical image segmentation. NAS U–Net [36] as a RL–based model is also considered for comparison. NAS U–Net is a 2D network; therefore, we converted 2D NAS U–Net to 3D according to our proposed model. All four networks were implemented and trained based on their reference papers. The number of training images and batch size were set as 32000 and 16 respectively, the same as our proposed model. All four networks were trained for five epochs and their DSCs are reported in Table VIII. As shown in Table VIII, in all cases our proposed converted 3D models outperformed these prior works for segmentation of the nine different datasets. The 3D U–Net method [57], in some cases such as Liver and Hippocampus datasets obtained high performance, however, in other instances it obtained almost average accuracy. Besides, Convnet [58] achieved nearly the highest segmentation accuracy for segmentation of the Liver and Spleen datasets, however in some cases such as CHAOS and Hippocampus, obtained very low efficiency. Also, 3D Dense U–Net [59] on average got the worst results and for all cases achieved an average or low accuracy. The reason for this is that all these networks are manually developed for a specific application or dataset. Therefore, their functionality is acceptable in some cases, but they need to be fine–tuned or modified to be applicable to another application or dataset. Besides, NAS U–Net [36] obtained almost the similar results of the manual–designed networks, because, NAS U–Net also uses a unique network structure for segmentation. Since our proposed model develops a network based on the dataset automatically, we can see that it worked well in different datasets and in all cases outperformed the previous 3D networks.

Besides, if we compared the evolutionary 2D networks versus the previous 3D networks, we can see that the 2D networks obtained competitive results. For example, using evolutionary 2D networks for segmentation of the CHAOS, Liver, Pancreas, Hippocampus, and Heart segmentation in some cases, they even outperformed previous 3D networks. Generally, well designed 3D networks outperformed 2D networks because they could use contextual information for segmentation of 3D instances; however, we can see that a well–designed evolutionary 2D network can even exceed manually–designed 3D networks.

Furthermore, we compared the obtained networks using nine different datasets versus previous works in terms of the number of trainable parameters. As shown in Table IX, the obtained networks using the proposed model are far smaller than prior works. All obtained networks used less than ten million parameters, that means they need less computation and time for training. However, as discussed above; they are still able to obtain high accuracy for medical image segmentation.

TABLE IX
COMPARISON OF THE PROPOSED MODEL VERSUS PRIOR WORKS IN TERMS OF NUMBER OF TRAINABLE PARAMETERS

Model	Number of Parameters
3D U–Net [57]	19.0×10^6
Convnet [58]	23.4×10^6
3D Dense U–Net [59]	43.8×10^6
3D NAS U–Net [36]	46.8×10^6
Converted 3D Network (PROMISE)	9.8×10^6
Converted 3D Network (CHAOS)	4.3×10^6
Converted 3D Network (SLIVER07)	2.4×10^6
Converted 3D Network (LIVER)	4.0×10^6
Converted 3D Network (SPLEEN)	9.0×10^6
Converted 3D Network (PANCREAS)	6.6×10^6
Converted 3D Network (PROSTATE)	8.5×10^6
Converted 3D Network (HIPPOCAMPUS)	7.3×10^6
Converted 3D Network (HEART)	6.2×10^6

V. DISCUSSION AND CONCLUSION

In this section, we discuss why using converted evolutionary 3D networks can be even better than purely evolutionary 3D networks. As discussed above, to develop the 3D networks, we first developed evolutionary 2D networks. In our proposed evolutionary 2D method, the training starts using 60 networks and continued to another eight generations using 30 population, and each network was trained with 50000 image slices. Totally, during nine generations, 300 networks were evolved. In the case of the PROMISE12 dataset, training takes about 210 hours using four GPUs (on average 0.7 hours per network). Then the evolutionary 2D networks were converted to 3D networks in seconds. However, in the evolutionary 3D model, training was started using 12 population and continued to nine generations using the same population size, and each network was trained to utilise 2000 MRI volumes. Overall, in the 3D model, just 108 networks were evolved in about 293 hours using the same number of GPUs and CPUs (on average 2.7 hours per network). The provided information shows the substantial differences between the two strategies to develop 3D networks, in terms of time and required computation.

Suppose developing an evolutionary 3D model using the same population size of the 2D model, in spite of using a smaller number of training data; the processing time and required computation will be about four times that of the 2D model. In other words, using the evolutionary 2D model to evolve 3D models, we saved about 75% in required computation. Most importantly, the converted 3D networks worked well in 3D volume segmentation applications that shows using the evolutionary 2D model to develop 3D networks is a reliable strategy to develop 3D models.

Furthermore, there is no guarantee to find a well–structured 3D network using a higher number of population size, but utilising a small number of training images. Based on our preliminary experiments, using a higher population size can increase the chance of finding a better network structure. However, the number of training examples is also effective in the network's evolution, such that networks using a higher number of parameters need more training data.

For example, a good network structure with a high number of trainable parameters, which are trained by a limited number of training images will obtain low accuracy on the validation set. Consequently, it will be removed from the population during evolution. Therefore, the evolution will lead to picking smaller networks with almost average accuracy. This problem happens in our evolutionary 3D model. Where the networks using a lower number of trainable parameters obtained better results using 2000 training images, and the larger networks were defeated in the evolution's competition. Consequently, the ten best networks that use a small number of trainable parameters were not generalised enough for segmentation of the unseen volumes, which is a relatively complicated task.

The problem is that despite having substantial computation facilities; it will still be impossible to increase the number of population size and/or training images to develop a 3D evolutionary network. As the results show, the quality of the obtained 3D evolutionary networks will be reduced.

Besides, to show converting 2D networks to 3D networks is an effective method to design 3D networks, we examined eight more datasets. Our experimental results show that in regards to the difficulty of developing evolutionary 3D networks, using a well–defined evolutionary 2D model can be a successful substitution to develop 3D networks. Of course, some of the findings may be restricted to the search space, datasets, and the specific application that we used for evaluation. A direction for future work can be using more datasets and other applications such as classification, to verify the generality of the proposed model.

Overall, The proposed model outperformed the evolutionary 3D model using less computation and processing time and resulted in at least 75% saving in time and computation. Also, compared to manually–designed 3D networks, our proposed 3D networks obtained higher accuracy. The proposed model shows that the 2D image slices corresponding to 3D volumes are informative enough to be used to develop the initial 2D networks, before they are converted to 3D networks.

REFERENCES

[1] Y. LeCun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*. [Online]. Available: http://arxiv.org/abs/1412.7062
[4] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*. [Online]. Available: http://arxiv.org/abs/1207.0580
[5] J. D. Schaffer, D. Whitley, and L. J. Eshelman, "Combinations of genetic algorithms and neural networks: A survey of the state of the art," in *Proc. Int. Workshop Combinations Genetic Algorithms Neural Netw. (COGANN)*, 1992, pp. 1–37.
[6] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Mach. Intell.*, vol. 1, no. 1, pp. 24–35, Jan. 2019.
[7] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: From architectures to learning," *Evol. Intell.*, vol. 1, no. 1, pp. 47–62, Mar. 2008.
[8] A. Baldominos, Y. Saez, and P. Isasi, "On the automated, evolutionary design of neural networks: Past, present, and future," *Neural Comput. Appl.*, vol. 32, pp. 1–27, Jan. 2020.
[9] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 97–116, Apr. 2017.

[10] X. Yao and M. M. Islam, "Evolving artificial neural network ensembles," *IEEE Comput. Intell. Mag.*, vol. 3, no. 1, pp. 31–42, Feb. 2008.

[11] D. Reinsel, J. Gantz, and J. Rydning, "Data age 2025: The evolution of data to life-critical don't focus on big data," *Framingham, IDC Analyze Future*, 2017.

[12] S. Medicine. *Health Trends Report: Harnessing the Power of Data in Health*. Accessed: 2017. [Online]. Available: https://med.stanford.edu/content/dam/sm/sm-news/documents/StanfordMedicineHealthTrendsWhitePaper2017.pdf

[13] H. LANDI. *IBM Unveils Watson-Powered Imaging Solutions at RSNA*. Accessed: 2016. [Online]. Available: https://www.hcinnovationgroup.com/population-health-management/news/13027814/ibm-unveils-watsonpowered-imaging-solutions-at-rsna

[14] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," in *Proc. IJCAI*, vol. 89, 1989, pp. 762–767.

[15] J. Holland, "Adaptation in natural and artificial systems: An introductory analysis with application to biology," in *Control and Artificial Intelligence*. Ann Arbor, MI, USA: Univ. of Michigan Press, 1975.

[16] G. F. Miller, P. M. Todd, and S. U. Hegde, "Designing neural networks using genetic algorithms," in *Proc. ICGA*, vol. 89, 1989, pp. 379–384.

[17] J. Koutník, J. Schmidhuber, and F. Gomez, "Evolving deep unsupervised convolutional networks for vision-based reinforcement learning," in *Proc. Conf. Genetic Evol. Comput. (GECCO)*, 2014, pp. 541–548.

[18] R. Miikkulainen *et al.*, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 293–312.

[19] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, Jun. 2002.

[20] A. Martín, R. Lara-Cabrera, F. Fuentes-Hurtado, V. Naranjo, and D. Camacho, "EvoDeep: A new evolutionary approach for automatic deep neural networks parametrisation," *J. Parallel Distrib. Comput.*, vol. 117, pp. 180–191, Jul. 2018.

[21] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, Apr. 2020.

[22] M. J. Shafiee, A. Mishra, and A. Wong, "Deep learning with Darwin: Evolutionary synthesis of deep neural networks," *Neural Process. Lett.*, vol. 48, no. 1, pp. 603–613, Aug. 2018.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[24] A. Baldominos, Y. Saez, and P. Isasi, "Evolutionary convolutional neural networks: An application to handwriting recognition," *Neurocomputing*, vol. 283, pp. 38–52, Mar. 2018.

[25] C. Ryan, J. J. Collins, and M. O. Neill, "Grammatical evolution: Evolving programs for an arbitrary language," in *Proc. Eur. Conf. Genetic Program.* Berlin, Germany: Springer, 1998, pp. 83–96.

[26] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948.

[28] B. Wang, Y. Sun, B. Xue, and M. Zhang, "A hybrid differential evolution approach to designing deep convolutional neural networks for image classification," in *Proc. Australas. Joint Conf. Artif. Intell.* Cham, Switzerland: Springer, 2018, pp. 237–250.

[29] R. S. Sutton, A. G. Barto, *Introduction to Reinforcement Learning*, vol. 2, no. 4. Cambridge, MA, USA: MIT Press, 1998.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[31] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*. [Online]. Available: http://arxiv.org/abs/1611.01578

[32] M. Wistuba, A. Rawat, and T. Pedapati, "A survey on neural architecture search," 2019, *arXiv:1905.01392*. [Online]. Available: http://arxiv.org/abs/1905.01392

[33] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[34] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4780–4789.

[35] N. Dong, M. Xu, X. Liang, Y. Jiang, W. Dai, and E. Xing, "Neural architecture search for adversarial medical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2019, pp. 828–836.

[36] Y. Weng, T. Zhou, Y. Li, and X. Qiu, "NAS-unet: Neural architecture search for medical image segmentation," *IEEE Access*, vol. 7, pp. 44247–44257, 2019.

[37] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[38] A. Mortazi and U. Bagci, "Automatically designing CNN architectures for medical image segmentation," in *Proc. Int. Workshop Mach. Learn. Med. Imag.* Cham, Switzerland: Springer, 2018, pp. 98–106.

[39] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 3, 2004, pp. 2619–2624.

[40] J. Liang, E. Meyerson, B. Hodjat, D. Fink, K. Mutch, and R. Miikkulainen, "Evolutionary neural AutoML for deep learning," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2019, pp. 13–17.

[41] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," 2018, *arXiv:1806.09055*. [Online]. Available: http://arxiv.org/abs/1806.09055

[42] R. Luo, F. Tian, T. Qin, E. Chen, and T.-Y. Liu, "Neural architecture optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7816–7827.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[44] R. Kumar Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, *arXiv:1505.00387*. [Online]. Available: http://arxiv.org/abs/1505.00387

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[46] T. Blickle and L. Thiele, "A comparison of selection schemes used in genetic algorithms. TIK-report 11, TIK institut fur technische informatik und kommunikationsnetze," Comput. Eng. Netw. Lab., ETH, Swiss Federal Inst. Technol., Zürich, Switzerland, 1995, p. 8092, vol. 35.

[47] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[48] D. Zill, W. S. Wright, and M. R. Cullen, *Advanced Engineering Mathematics*. Burlington, MA, USA: Jones & Bartlett Learning, 2011.

[49] T. Hassanzadeh, L. G. C. Hamey, and K. Ho-Shon, "Convolutional neural networks for prostate magnetic resonance image segmentation," *IEEE Access*, vol. 7, pp. 36748–36760, 2019.

[50] P. Y. Simard *et al.*, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. ICDAR*, vol. 3, no. 2003, 2003, pp. 958–963.

[51] F. Chollet *et al.*, "Keras: Deep learning library for theano and tensorflow," vol. 7, no. 8, p. T1, 2015. [Online]. Available: https://keras.io/k

[52] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945.

[53] T. Heimann *et al.*, "Comparison and evaluation of methods for liver segmentation from CT datasets," *IEEE Trans. Med. Imag.*, vol. 28, no. 8, pp. 1251–1265, Aug. 2009.

[54] S. S. Chandra *et al.*, "Patient specific prostate segmentation in 3-D magnetic resonance images," *IEEE Trans. Med. Imag.*, vol. 31, no. 10, pp. 1955–1964, Oct. 2012.

[55] A. E. Kavur, M. A. Selver, O. Dicle, M. Barış, and N. S. Gezer, *CHAOS—Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge Data, v1.03*. Zenodo, Apr. 2019, doi: 10.5281/zenodo.3362844.

[56] A. Simpson *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," 2019, *arXiv:1902.09063*. [Online]. Available: https://arxiv.org/abs/1902.09063

[57] O. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2016, pp. 424–432.

[58] L. Yu, X. Yang, H. Chen, J. Qin, and P. A. Heng, "Volumetric convnets with mixed residual connections for automated prostate segmentation from 3D MR images," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 36–72.

[59] M. Kolařík, R. Burget, V. Uher, K. Říha, and M. Dutta, "Optimized high resolution 3D Dense-U-Net network for brain and spine segmentation," *Appl. Sci.*, vol. 9, no. 3, p. 404, Jan. 2019.

nature communications

Article

https://doi.org/10.1038/s41467-024-44824-z

Segment anything in medical images

Received: 24 October 2023

Accepted: 5 January 2024

Published online: 22 January 2024

Check for updates

Jun Ma[1,2,3], Yuting He[4], Feifei Li[1], Lin Han[5], Chenyu You[6] & Bo Wang[1,2,3,7,8]

Medical image segmentation is a critical component in clinical practice, facilitating accurate diagnosis, treatment planning, and disease monitoring. However, existing methods, often tailored to specific modalities or disease types, lack generalizability across the diverse spectrum of medical image segmentation tasks. Here we present MedSAM, a foundation model designed for bridging this gap by enabling universal medical image segmentation. The model is developed on a large-scale medical image dataset with 1,570,263 image-mask pairs, covering 10 imaging modalities and over 30 cancer types. We conduct a comprehensive evaluation on 86 internal validation tasks and 60 external validation tasks, demonstrating better accuracy and robustness than modality-wise specialist models. By delivering accurate and efficient segmentation across a wide spectrum of tasks, MedSAM holds significant potential to expedite the evolution of diagnostic tools and the personalization of treatment plans.

Segmentation is a fundamental task in medical imaging analysis, which involves identifying and delineating regions of interest (ROI) in various medical images, such as organs, lesions, and tissues[1]. Accurate segmentation is essential for many clinical applications, including disease diagnosis, treatment planning, and monitoring of disease progression[2,3]. Manual segmentation has long been the gold standard for delineating anatomical structures and pathological regions, but this process is time-consuming, labor-intensive, and often requires a high degree of expertise. Semi- or fully automatic segmentation methods can significantly reduce the time and labor required, increase consistency, and enable the analysis of large-scale datasets[4].

Deep learning-based models have shown great promise in medical image segmentation due to their ability to learn intricate image features and deliver accurate segmentation results across a diverse range of tasks, from segmenting specific anatomical structures to identifying pathological regions[5]. However, a significant limitation of many current medical image segmentation models is their task-specific nature. These models are typically designed and trained for a specific segmentation task, and their performance can degrade significantly when applied to new tasks or different types of imaging data[6]. This lack of generality poses a substantial obstacle to the wider application of these models in clinical practice. In contrast, recent advances in the field of natural image segmentation have witnessed the emergence of segmentation foundation models, such as segment anything model (SAM)[7] and Segment Everything Everywhere with Multi-modal prompts all at once[8], showcasing remarkable versatility and performance across various segmentation tasks.

There is a growing demand for universal models in medical image segmentation: models that can be trained once and then applied to a wide range of segmentation tasks. Such models would not only exhibit heightened versatility in terms of model capacity but also potentially lead to more consistent results across different tasks. However, the applicability of the segmentation foundation models (e.g., SAM[7]) to medical image segmentation remains limited due to the significant differences between natural images and medical images. Essentially, SAM is a promptable segmentation method that requires points or bounding boxes to specify the segmentation targets. This resembles conventional interactive segmentation methods[4,9–11] but SAM has better generalization ability, while existing deep learning-based interactive segmentation methods focus mainly on limited tasks and image modalities.

Many studies have applied the out-of-the-box SAM models to typical medical image segmentation tasks[12–17] and other challenging scenarios[18–21]. For example, the concurrent studies[22,23] conducted a

[1]Peter Munk Cardiac Centre, University Health Network, Toronto, ON, Canada. [2]Department of Laboratory Medicine and Pathobiology, University of Toronto, Toronto, ON, Canada. [3]Vector Institute, Toronto, ON, Canada. [4]Department of Computer Science, Western University, London, ON, Canada. [5]Tandon School of Engineering, New York University, New York, NY, USA. [6]Department of Electrical Engineering, Yale University, New Haven, CT, USA. [7]Department of Computer Science, University of Toronto, Toronto, ON, Canada. [8]UHN AI Hub, Toronto, ON, Canada. e-mail: bowang@vectorinstitute.ai

comprehensive assessment of SAM across a diverse array of medical images, underscoring that SAM achieved satisfactory segmentation outcomes primarily on targets characterized by distinct boundaries. However, the model exhibited substantial limitations in segmenting typical medical targets with weak boundaries or low contrast. In congruence with these observations, we further introduce MedSAM, a refined foundation model that significantly enhances the segmentation performance of SAM on medical images. MedSAM accomplishes this by fine-tuning SAM on an unprecedented dataset with more than one million medical image-mask pairs.

We thoroughly evaluate MedSAM through comprehensive experiments on 86 internal validation tasks and 60 external validation tasks, spanning a variety of anatomical structures, pathological conditions, and medical imaging modalities. Experimental results demonstrate that MedSAM consistently outperforms the state-of-the-art (SOTA) segmentation foundation model[7], while achieving performance on par with, or even surpassing specialist models[1,24] that were trained on the images from the same modality. These results highlight the potential of MedSAM as a new paradigm for versatile medical image segmentation.

Results

MedSAM: a foundation model for promptable medical image segmentation

MedSAM aims to fulfill the role of a foundation model for universal medical image segmentation. A crucial aspect of constructing such a model is the capacity to accommodate a wide range of variations in imaging conditions, anatomical structures, and pathological conditions. To address this challenge, we curated a diverse and large-scale medical image segmentation dataset with 1,570,263 medical image-mask pairs, covering 10 imaging modalities, over 30 cancer types, and a multitude of imaging protocols (Fig. 1 and Supplementary Tables 1–4). This large-scale dataset allows MedSAM to learn a rich representation of medical images, capturing a broad spectrum of anatomies and lesions across different modalities. Figure 2a provides an overview of the distribution of images across different medical imaging modalities in the dataset, ranked by their total numbers. It is evident that computed tomography (CT), magnetic resonance imaging (MRI), and endoscopy are the dominant modalities, reflecting their ubiquity in clinical practice. CT and MRI images provide detailed cross-sectional views of 3D body structures, making them indispensable for non-invasive diagnostic imaging. Endoscopy, albeit more invasive, enables direct visual inspection of organ interiors, proving invaluable for diagnosing gastrointestinal and urological conditions. Despite the prevalence of these modalities, others such as ultrasound, pathology, fundus, dermoscopy, mammography, and optical coherence tomography (OCT) also hold significant roles in clinical practice. The diversity of these modalities and their corresponding segmentation targets underscores the necessity for universal and effective segmentation models capable of handling the unique characteristics associated with each modality.

Another critical consideration is the selection of the appropriate segmentation prompt and network architecture. While the concept of fully automatic segmentation foundation models is enticing, it is fraught with challenges that make it impractical. One of the primary challenges is the variability inherent in segmentation tasks. For example, given a liver cancer CT image, the segmentation task can vary depending on the specific clinical scenario. One clinician might be

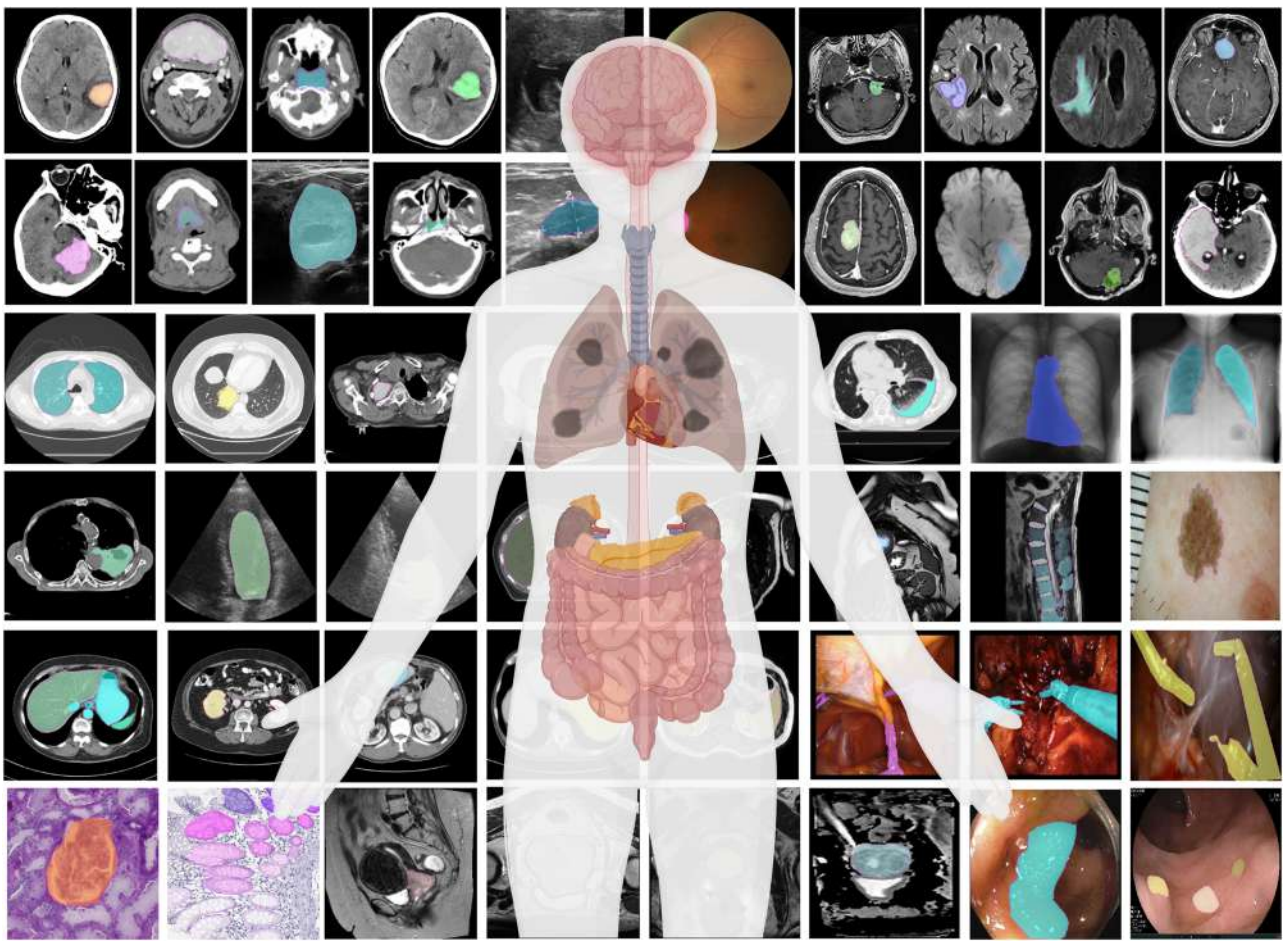


Fig. 1 | MedSAM is trained on a large-scale dataset that can handle diverse segmentation tasks. The dataset covers a variety of anatomical structures, pathological conditions, and medical imaging modalities. The magenta contours and mask overlays denote the expert annotations and MedSAM segmentation results, respectively.

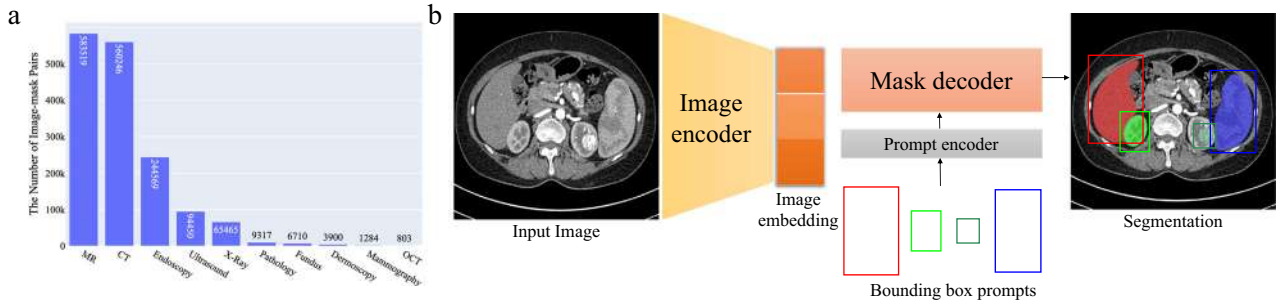


Fig. 2 | Overview of the modality distribution in the dataset and the network architecture. a The number of medical image-mask pairs in each modality. **b** MedSAM is a promptable segmentation method where users can use bounding boxes to specify the segmentation targets. Source data are provided as a Source Data file.

interested in segmenting the liver tumor, while another might need to segment the entire liver and surrounding organs. Additionally, the variability in imaging modalities presents another challenge. Modalities such as CT and MR generate 3D images, whereas others like X-ray and ultrasound yield 2D images. These variabilities in task definition and imaging modalities complicate the design of a fully automatic model capable of accurately anticipating and addressing the diverse requirements of different users.

Considering these challenges, we argue that a more practical approach is to develop a promptable 2D segmentation model. The model can be easily adapted to specific tasks based on user-provided prompts, offering enhanced flexibility and adaptability. It is also able to handle both 2D and 3D images by processing 3D images as a series of 2D slices. Typical user prompts include points and bounding boxes and we show some segmentation examples with the different prompts in Supplementary Fig. 1. It can be found that bounding boxes provide a more unambiguous spatial context for the region of interest, enabling the algorithm to more precisely discern the target area. This stands in contrast to point-based prompts, which can introduce ambiguity, particularly when proximate structures resemble each other. Moreover, drawing a bounding box is efficient, especially in scenarios involving multi-object segmentation. We follow the network architecture in SAM[7], including an image encoder, a prompt encoder, and a mask decoder (Fig. 2b). The image encoder[25] maps the input image into a high-dimensional image embedding space. The prompt encoder transforms the user-drawn bounding boxes into feature representations via positional encoding[26]. Finally, the mask decoder fuses the image embedding and prompt features using cross-attention[27] (Methods).

Quantitative and qualitative analysis

We evaluated MedSAM through both internal validation and external validation. Specifically, we compared it to the SOTA segmentation foundation model SAM[7] as well as modality-wise specialist U-Net[1] and DeepLabV3+[24] models. Each specialized model was trained on images from the corresponding modality, resulting in 10 dedicated specialist models for each method. During inference, these specialist models were used to segment the images from corresponding modalities, while SAM and MedSAM were employed for segmenting images across all modalities (Methods). The internal validation contained 86 segmentation tasks (Supplementary Tables 5–8 and Fig. 2), and Fig. 3a shows the median dice similarity coefficient (DSC) score of these tasks for the four methods. Overall, SAM obtained the lowest performance on most segmentation tasks although it performed promisingly on some RGB image segmentation tasks, such as polyp (DSC: 91.3%, interquartile range (IQR): 81.2–95.1%) segmentation in endoscopy images. This could be attributed to SAM's training on a variety of RGB images, and the fact that many targets in these images are relatively straightforward to segment due to their distinct appearances. The other three models outperformed SAM by a large margin and MedSAM has a narrower distribution of DSC scores of the 86 interval validation tasks than the two groups of specialist models, reflecting the robustness of MedSAM across different tasks. We further connected the DSC scores corresponding to the same task of the four models with the podium plot Fig. 3b, which is complementary to the box plot. In the upper part, each colored dot denotes the median DSC achieved with the respective method on one task. Dots corresponding to identical test cases are connected by a line. In the lower part, the frequency of achieved ranks for each method is presented with bar charts. It can be found that MedSAM ranked in first place on most tasks, surpassing the performance of the U-Net and DeepLabV3+ specialist models that have a high frequency of ranks with second and third places, respectively, In contrast, SAM ranked last place in almost all tasks. Figure 3c (and Supplementary Fig. 9) visualizes some randomly selected segmentation examples where MedSAM obtained a median DSC score, including liver tumor in CT images, brain tumor in MR images, breast tumor in ultrasound images, and polyp in endoscopy images. SAM struggles with targets of weak boundaries, which is prone to under or over-segmentation errors. In contrast, MedSAM can accurately segment a wide range of targets across various imaging conditions, which achieves comparable of even better than the specialist U-Net and DeepLabV3+ models.

The external validation included 60 segmentation tasks, all of which either were from new datasets or involved unseen segmentation targets (Supplementary Tables 9–11 and Figs. 10–12). Figure 4a, b show the task-wise median DSC score distribution and their correspondence of the 60 tasks, respectively. Although SAM continued exhibiting lower performance on most CT and MR segmentation tasks, the specialist models no longer consistently outperformed SAM (e.g., right kidney segmentation in MR T1-weighted images: 90.1%, 85.3%, 86.4% for SAM, U-Net, and DeepLabV3+, respectively). This indicates the limited generalization ability of such specialist models on unseen targets. In contrast, MedSAM consistently delivers superior performance. For example, MedSAM obtained median DSC scores of 87.8% (IQR: 85.0-91.4%) on the nasopharynx cancer segmentation task, demonstrating 52.3%, 15.5%, and 22.7 improvements over SAM, the specialist U-Net, and DeepLabV3+, respectively. Significantly, MedSAM also achieved better performance in some unseen modalities (e.g., abdomen T1 Inphase and Outphase), surpassing SAM and the specialist models with improvements by up to 10%. Figure 4c presents four randomly selected segmentation examples for qualitative evaluation, revealing that while all the methods have the ability to handle simple segmentation targets, MedSAM performs better at segmenting challenging targets with indistinguishable boundaries, such as cervical cancer in MR images (more examples are presented in Supplementary Fig. 13). Furthermore, we evaluated MedSAM on the multiple myeloma plasma cell dataset, which represents a distinct modality and task in contrast to all previously leveraged validation tasks. Although this task had never been seen during training,

https://doi.org/10.1038/s41467-024-44824-z

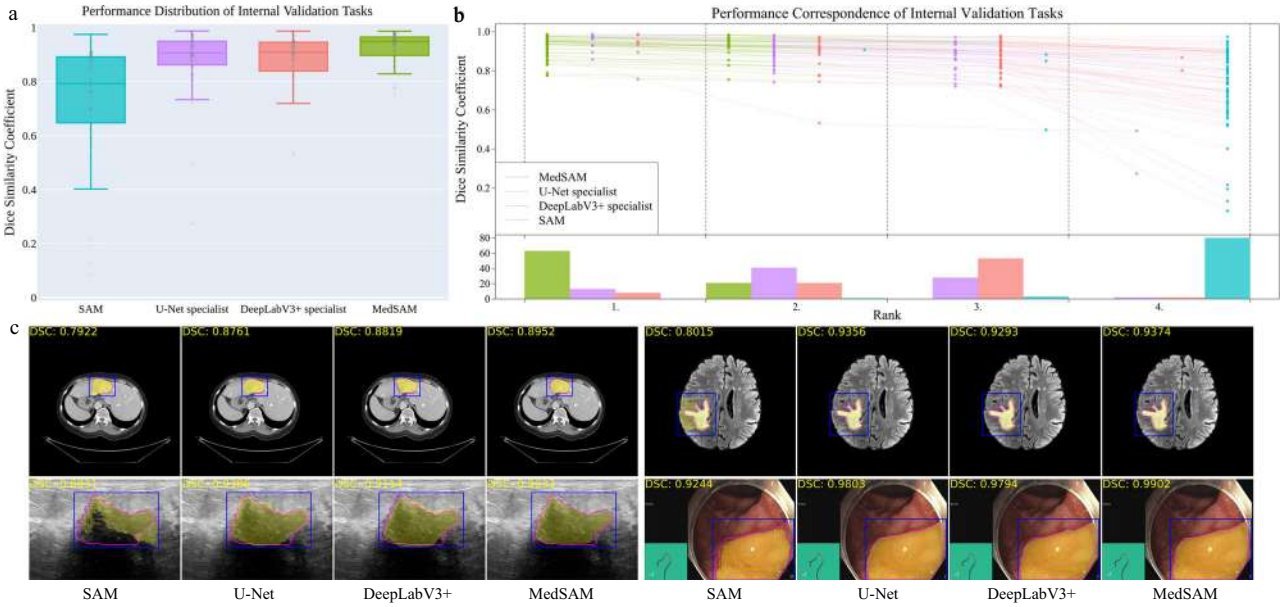


Fig. 3 | Quantitative and qualitative evaluation results on the internal validation set. a Performance distribution of 86 internal validation tasks in terms of median dice similarity coefficient (DSC) score. The center line within the box represents the median value, with the bottom and top bounds of the box delineating the 25th and 75th percentiles, respectively. Whiskers are chosen to show the 1.5 of the interquartile range. Up-triangles denote the minima and down-triangles denote the maxima. **b** Podium plots for visualizing the performance correspondence of 86 internal validation tasks. Upper part: each colored dot denotes the median DSC achieved with the respective method on one task. Dots corresponding to identical tasks are connected by a line. Lower part: bar charts represent the frequency of achieved ranks for each method. MedSAM ranks in the first place on most tasks. **c** Visualized segmentation examples on the internal validation set. The four examples are liver cancer, brain cancer, breast cancer, and polyp in computed tomography (CT), (Magnetic Resonance Imaging) MRI, ultrasound, and endoscopy images, respectively. Blue: bounding box prompts; Yellow: segmentation results. Magenta: expert annotations. Source data are provided as a Source Data file.

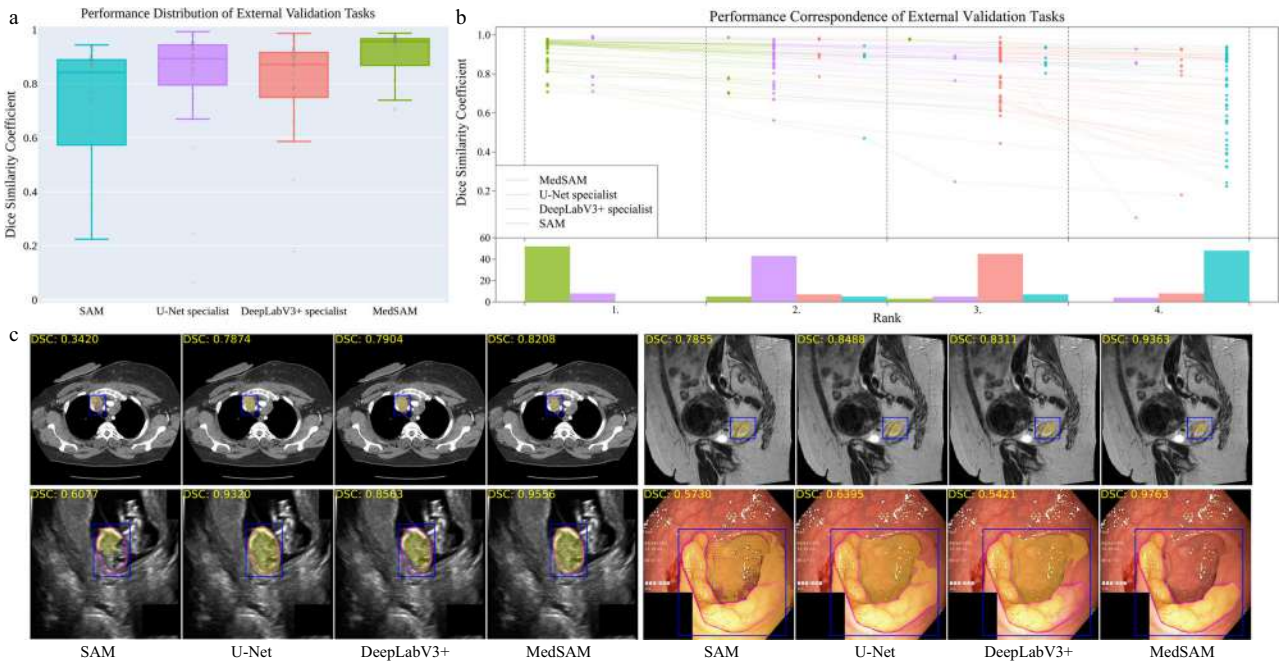


Fig. 4 | Quantitative and qualitative evaluation results on the external validation set. a Performance distribution of 60 external validation tasks in terms of median dice similarity coefficient (DSC) score. The center line within the box represents the median value, with the bottom and top bounds of the box delineating the 25th and 75th percentiles, respectively. Whiskers are chosen to show the 1.5 of the interquartile range. Up-triangles denote the minima and down-triangles denote the maxima. **b** Podium plots for visualizing the performance correspondence of 60 external validation tasks. Upper part: each colored dot denotes the median DSC achieved with the respective method on one task. Dots corresponding to identical tasks are connected by a line. Lower part: bar charts represent the frequency of achieved ranks for each method. MedSAM ranks in the first place on most tasks. **c** Visualized segmentation examples on the external validation set. The four examples are the lymph node, cervical cancer, fetal head, and polyp in CT, MR, ultrasound, and endoscopy images, respectively. Source data are provided as a Source Data file.

MedSAM still exhibited superior performance compared to the SAM (Supplementary Fig. 14), highlighting its remarkable generalization ability.

The effect of training dataset size

We also investigated the effect of varying dataset sizes on MedSAM's performance because the training dataset size has been proven to be pivotal in model performance[28]. We additionally trained MedSAM on two different dataset sizes: 10,000 (10K) and 100,000 (100K) images and their performances were compared with the default MedSAM model. The 10K and 100K training images were uniformly sampled from the whole training set, to maintain data diversity. As shown in (Fig. 5a) (Supplementary Tables 12–14), the performance adhered to the scaling rule, where increasing the number of training images significantly improved the performance in both internal and external validation sets.

MedSAM can improve the annotation efficiency

Furthermore, we conducted a human annotation study to assess the time cost of two pipelines (Methods). For the first pipeline, two human experts manually annotate 3D adrenal tumors in a slice-by-slice way. For the second pipeline, the experts first drew the long and short tumor axes with the linear marker (initial marker) every 3-10 slices, which is a common practice in tumor response evaluation. Then, MedSAM was used to segment the tumors based on these sparse linear annotations. Finally, the expert manually revised the segmentation results until they were satisfied. We quantitatively compared the annotation time cost between the two pipelines (Fig. 5b). The results demonstrate that with the assistance of MedSAM, the annotation time is substantially reduced by 82.37% and 82.95% for the two experts, respectively.

Discussion

We introduce MedSAM, a deep learning-powered foundation model designed for the segmentation of a wide array of anatomical structures and lesions across diverse medical imaging modalities. MedSAM is trained on a meticulously assembled large-scale dataset comprised of over one million medical image-mask pairs. Its promptable configuration strikes an optimal balance between automation and customization, rendering MedSAM a versatile tool for universal medical image segmentation.

Through comprehensive evaluations encompassing both internal and external validation, MedSAM has demonstrated substantial capabilities in segmenting a diverse array of targets and robust generalization abilities to manage new data and tasks. Its performance not only significantly exceeds that of existing the state-of-the-art segmentation foundation model, but also rivals or even surpasses specialist models. By providing precise delineation of anatomical structures and pathological regions, MedSAM facilitates the computation of various quantitative measures that serve as biomarkers. For instance, in the field of oncology, MedSAM could play a crucial role in accelerating the 3D tumor annotation process, enabling subsequent calculations of tumor volume, which is a critical biomarker[29] for assessing disease progression and response to treatment. Additionally, MedSAM provides a successful paradigm for adapting natural image foundation models to new domains, which can be further extended to biological image segmentation[30], such as cell segmentation in light microscopy images[31] and organelle segmentation in electron microscopy images[32].

While MedSAM boasts strong capabilities, it does present certain limitations. One such limitation is the modality imbalance in the training set, with CT, MRI, and endoscopy images dominating the dataset. This could potentially impact the model's performance on less-represented modalities, such as mammography. Another limitation is its difficulty in the segmentation of vessel-like branching structures because the bounding box prompt can be ambiguous in this setting. For example, arteries and veins share the same bounding box in eye fundus images. However, these limitations do not diminish MedSAM's utility. Since MedSAM has learned rich and representative medical image features from the large-scale training set, it can be fine-tuned to effectively segment new tasks from less-represented modalities or intricate structures like vessels.

In conclusion, this study highlights the feasibility of constructing a single foundation model capable of managing a multitude of segmentation tasks, thereby eliminating the need for task-specific models. MedSAM, as the inaugural foundation model in medical image segmentation, holds great potential to accelerate the advancement of new diagnostic and therapeutic tools, and ultimately contribute to improved patient care[33].

Methods

Dataset curation and pre-processing

We curated a comprehensive dataset by collating images from publicly available medical image segmentation datasets, which were obtained from various sources across the internet, including the Cancer Imaging Archive (TCIA)[34], Kaggle, Grand-Challenge, Scientific Data, CodaLab, and segmentation challenges in the Medical Image Computing and Computer Assisted Intervention Society (MICCAI). All the datasets provided segmentation annotations by human experts, which have been widely used in existing literature (Supplementary Table 1–4). We incorporated these annotations directly for both model development and validation.

The original 3D datasets consisted of computed tomography (CT) and magnetic resonance (MR) images in DICOM, nrrd, or mhd formats. To ensure uniformity and compatibility with developing medical image deep learning models, we converted the images to the widely used NifTI format. Additionally, grayscale images (such as X-Ray and Ultrasound) as well as RGB images (including endoscopy, dermoscopy, fundus, and pathology images), were converted to the png format.

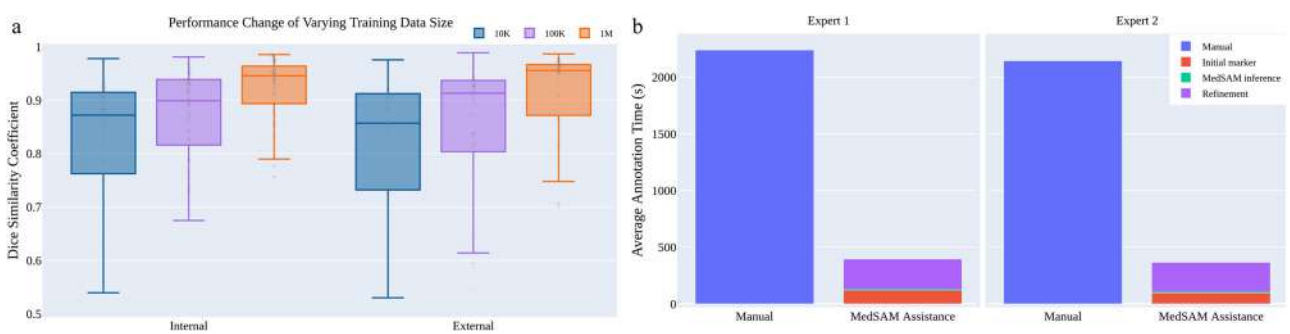


Fig. 5 | The effect of training dataset size and a user study of tumor annotation efficiency. a Scaling up the training image size to one million can significantly improve the model performance on both internal and external validation sets. **b** MedSAM can be used to substantially reduce the annotation time cost. Source data are provided as a Source Data file.

Several exclusive criteria are applied to improve the dataset quality and consistency, including incomplete images and segmentation targets with branching structures, inaccurate annotations, and tiny volumes. Notably, image intensities varied significantly across different modalities. For instance, CT images had intensity values ranging from -2000 to 2000, while MR images exhibited a range of 0 to 3000. In endoscopy and ultrasound images, intensity values typically spanned from 0 to 255. To facilitate stable training, we performed intensity normalization across all images, ensuring they shared the same intensity range.

For CT images, we initially normalized the Hounsfield units using typical window width and level values. The employed window width and level values for soft tissues, lung, and brain are (W:400, L:40), (W:1500, L:-160), and (W:80, L:40), respectively. Subsequently, the intensity values were rescaled to the range of [0, 255]. For MR, X-ray, ultrasound, mammography, and optical coherence tomography (OCT) images, we clipped the intensity values to the range between the 0.5th and 99.5th percentiles before rescaling them to the range of [0, 255]. Regarding RGB images (e.g., endoscopy, dermoscopy, fundus, and pathology images), if they were already within the expected intensity range of [0, 255], their intensities remained unchanged. However, if they fell outside this range, we utilized max-min normalization to rescale the intensity values to [0, 255]. Finally, to meet the model's input requirements, all images were resized to a uniform size of $1024 \times 1024 \times 3$. In the case of whole-slide pathology images, patches were extracted using a sliding window approach without overlaps. The patches located on boundaries were padded to this size with 0. As for 3D CT and MR images, each 2D slice was resized to 1024×1024, and the channel was repeated three times to maintain consistency. The remaining 2D images were directly resized to $1024 \times 1024 \times 3$. Bi-cubic interpolation was used for resizing images, while nearest-neighbor interpolation was applied for resizing masks to preserve their precise boundaries and avoid introducing unwanted artifacts. These standardization procedures ensured uniformity and compatibility across all images and facilitated seamless integration into the subsequent stages of the model training and evaluation pipeline.

Network architecture

The network utilized in this study was built on transformer architecture[27], which has demonstrated remarkable effectiveness in various domains such as natural language processing and image recognition tasks[25]. Specifically, the network incorporated a vision transformer (ViT)-based image encoder responsible for extracting image features, a prompt encoder for integrating user interactions (bounding boxes), and a mask decoder that generated segmentation results and confidence scores using the image embedding, prompt embedding, and output token.

To strike a balance between segmentation performance and computational efficiency, we employed the base ViT model as the image encoder since extensive evaluation indicated that larger ViT models, such as ViT Large and ViT Huge, offered only marginal improvements in accuracy[7] while significantly increasing computational demands. Specifically, the base ViT model consists of 12 transformer layers[27], with each block comprising a multi-head self-attention block and a Multilayer Perceptron (MLP) block incorporating layer normalization[35]. Pre-training was performed using masked auto-encoder modeling[36], followed by fully supervised training on the SAM dataset[7]. The input image ($1024 \times 1024 \times 3$) was reshaped into a sequence of flattened 2D patches with the size $16 \times 16 \times 3$, yielding a feature size in image embedding of 64×64 after passing through the image encoder, which is $16 \times$ downscaled. The prompt encoders mapped the corner point of the bounding box prompt to 256-dimensional vectorial embeddings[26]. In particular, each bounding box was represented by an embedding pair of the top-left corner point and the bottom-right corner point. To facilitate real-time user interactions once the image embedding had been computed, a lightweight mask decoder architecture was employed. It consists of two transformer layers[27] for fusing the image embedding and prompt encoding, and two transposed convolutional layers to enhance the embedding resolution to 256×256. Subsequently, the embedding underwent sigmoid activation, followed by bi-linear interpolations to match the input size.

Training protocol and experimental setting

During data pre-processing, we obtained 1,570,263 medical image-mask pairs for model development and validation. For internal validation, we randomly split the dataset into 80%, 10%, and 10% as training, tuning, and validation, respectively. Specifically, for modalities where within-scan continuity exists, such as CT and MRI, and modalities where continuity exists between consecutive frames, we performed the data splitting at the 3D scan and the video level respectively, by which any potential data leak was prevented. For pathology images, recognizing the significance of slide-level cohesiveness, we first separated the whole-slide images into distinct slide-based sets. Then, each slide was divided into small patches with a fixed size of 1024×1024. This setup allowed us to monitor the model's performance on the tuning set and adjust its parameters during training to prevent overfitting. For the external validation, all datasets were held out and did not appear during model training. These datasets provide a stringent test of the model's generalization ability, as they represent new patients, imaging conditions, and potentially new segmentation tasks that the model has not encountered before. By evaluating the performance of MedSAM on these unseen datasets, we can gain a realistic understanding of how MedSAM is likely to perform in real-world clinical settings, where it will need to handle a wide range of variability and unpredictability in the data. The training and validation are independent.

The model was initialized with the pre-trained SAM model with the ViT-Base model. We fixed the prompt encoder since it can already encode the bounding box prompt. All the trainable parameters in the image encoder and mask decoder were updated during training. Specifically, the number of trainable parameters for the image encoder and mask decoder are 89,670,912 and 4,058,340, respectively. The bounding box prompt was simulated from the expert annotations with a random perturbation of 0-20 pixels. The loss function is the unweighted sum between dice loss and cross-entropy loss, which has been proven to be robust in various segmentation tasks[1]. The network was optimized by AdamW[37] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with an initial learning rate of 1e-4 and a weight decay of 0.01. The global batch size was 160 and data augmentation was not used. The model was trained on 20 A100 (80G) GPUs with 150 epochs and the last checkpoint was selected as the final model.

Furthermore, to thoroughly evaluate the performance of MedSAM, we conducted comparative analyses against both the state-of-the-art segmentation foundation model SAM[7] and specialist models (i.e., U-Net[1] and DeepLabV3+[24]). The training images contained 10 modalities: CT, MR, chest X-ray (CXR), dermoscopy, endoscopy, ultrasound, mammography, OCT, and pathology, and we trained the U-Net and DeepLabV3+ specialist models for each modality. There were 20 specialist models in total and the number of corresponding training images was presented in Supplementary Table 5. We employed the nnU-Net to conduct all U-Net experiments, which can automatically configure the network architecture based on the dataset properties. In order to incorporate the bounding box prompt into the model, we transformed the bounding box into a binary mask and concatenated it with the image as the model input. This function was originally supported by nnU-Net in the cascaded pipeline, which has demonstrated increased performance in many segmentation tasks by using the binary mask as an additional channel to specify the target location. The training settings followed the default configurations of 2D nnU-Net. Each model was trained on one A100 GPU with 1000

epochs and the last checkpoint was used as the final model. The DeepLabV3+ specialist models used ResNet50[38] as the encoder. Similar to ref. 3, the input images were resized to 224 × 224 × 3. The bounding box was transformed into a binary mask as an additional input channel to provide the object location prompt. Segmentation Models Pytorch (0.3.3)[39] was used to perform training and inference for all the modality-wise specialist DeepLabV3 + models. Each modality-wise model was trained on one A100 GPU with 500 epochs and the last checkpoint was used as the final model. During the inference phase, SAM and MedSAM were used to perform segmentation across all modalities with a single model. In contrast, the U-Net and DeepLabV3+ specialist models were used to individually segment the respective corresponding modalities.

A task-specific segmentation model might outperform a modality-based one for certain applications. Since U-Net obtained better performance than DeepLabV3+ on most tasks, we further conducted a comparison study by training task-specific U-Net models on four representative tasks, including liver cancer segmentation in CT scans, abdominal organ segmentation in MR scans, nerve cancer segmentation in ultrasound, and polyp segmentation in endoscopy images. The experiments included both internal validation and external validation. For internal validation, we adhered to the default data splits, using them to train the task-specific U-Net models and then evaluate their performance on the corresponding validation set. For external validation, the trained U-Net models were evaluated on new datasets from the same modality or segmentation targets. In all these experiments, MedSAM was directly applied to the validation sets without additional fine-tuning. As shown in Supplementary Fig. 15, while task-specific U-Net models often achieved great results on internal validation sets, their performance diminished significantly for external sets. In contrast, MedSAM maintained consistent performance across both internal and external validation sets. This underscores MedSAM's superior generalization ability, making it a versatile tool in a variety of medical image segmentation tasks.

Loss function

We used the unweighted sum between cross-entropy loss and dice loss[40] as the final loss function since it has been proven to be robust across different medical image segmentation tasks[41]. Specifically, let S, G denote the segmentation result and ground truth, respectively. s_i, g_i denotes the predicted segmentation and ground truth of voxel i, respectively. N is the number of voxels in the image I. Binary cross-entropy loss is defined by

$$L_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[g_i \log s_i + (1-g_i)\,\log(1-s_i)\right], \tag{1}$$

and dice loss is defined by

$$L_{\mathrm{Dice}} = 1 - \frac{2\sum_{i=1}^{N} g_i s_i}{\sum_{i=1}^{N}(g_i)^2 + \sum_{i=1}^{N}(s_i)^2}. \tag{2}$$

The final loss L is defined by

$$L = L_{\mathrm{BCE}} + L_{\mathrm{Dice}}. \tag{3}$$

Human annotation study

The objective of the human annotation study was to quantitatively evaluate how MedSAM can reduce the annotation time cost. Specifically, we used the recent adrenocortical carcinoma CT dataset[34,42,43], where the segmentation target, adrenal tumor, was neither part of the training nor of the existing validation sets. We randomly sampled 10 cases, comprising a total of 733 tumor slices requiring annotations. Two human experts participated in this study, both of whom are experienced radiologists with 8 and 6 years of clinical practice in abdominal diseases, respectively. Each expert generated two groups of annotations, one with the assistance of MedSAM and one without.

In the first group, the experts manually annotated the 3D adrenal tumor in a slice-by-slice manner. Annotations by the two experts were conducted independently, with no collaborative discussions, and the time taken for each case was recorded. In the second group, annotations were generated after one week of cooling period. The experts independently drew the long and short tumor axes as initial markers, which is a common practice in tumor response evaluation. This process was executed every 3-10 slices from the top slice to the bottom slice of the tumor. Then, we applied MedSAM to segment the tumors based on these sparse linear annotations, including three steps.

- Step 1. For each annotated slice, a rectangle binary mask was generated based on the linear label that can completely cover the linear label.
- Step 2. For the unlabeled slices, the rectangle binary masks were created through interpolation of the surrounding labeled slices.
- Step 3. We transformed the binary masks into bounding boxes and then fed them along with the images into MedSAM to generate segmentation results.

All these steps were conducted in an automatic way and the model running time was recorded for each case. Finally, human experts manually refined the segmentation results until they met their satisfaction. To summarize, the time cost of the second group of annotations contained three parts: initial markers, MedSAM inference, and refinement. All the manual annotation processes were based on ITK-SNAP[44], an open-source software designed for medical image visualization and annotation.

Evaluation metrics

We followed the recommendations in Metrics Reloaded[45] and used the dice similarity coefficient and normalized surface distance (NSD) to quantitatively evaluate the segmentation results. DSC is a region-based segmentation metric, aiming to evaluate the region overlap between expert annotation masks and segmentation results, which is defined by

$$\mathrm{DSC}(G,S) = \frac{2|G \cap S|}{|G| + |S|},$$

NSD[46] is a boundary-based metric, aiming to evaluate the boundary consensus between expert annotation masks and segmentation results at a given tolerance, which is defined by

$$\mathrm{NSD}(G,S) = \frac{|\partial G \cap B_{\partial S}^{(\tau)}| + |\partial S \cap B_{\partial G}^{(\tau)}|}{|\partial G| + |\partial S|},$$

where $B_{\partial G}^{(\tau)} = \{x \in R^3 \mid \exists \tilde{x} \in \partial G, ||x - \tilde{x}|| \le \tau\}$, $B_{\partial S}^{(\tau)} = \{x \in R^3 \mid \exists \tilde{x} \in \partial S, ||x - \tilde{x}|| \le \tau\}$ denote the border region of the expert annotation mask and the segmentation surface at tolerance τ, respectively. In this paper, we set the tolerance τ as 2.

Statistical analysis

To statistically analyze and compare the performance of the aforementioned four methods (MedSAM, SAM, U-Net, and DeepLabV3+ specialist models), we employed the Wilcoxon signed-rank test. This non-parametric test is well-suited for comparing paired samples and is particularly useful when the data does not meet the assumptions of normal distribution. This analysis allowed us to determine if any method demonstrated statistically superior segmentation performance compared to the others, providing valuable insights into the comparative effectiveness of the evaluated methods. The Wilcoxon signed-rank test results are marked on the DSC and NSD score tables (Supplementary Table 6–11).

Software utilized

All code was implemented in Python (3.10) using Pytorch (2.0) as the base deep learning framework. We also used several Python packages for data analysis and results visualization, including connected-components-3d (3.10.3), SimpleITK (2.2.1), nibabel (5.1.0), torchvision (0.15.2), numpy (1.24.3), scikit-image (0.20.0), scipy (1.10.1), and pandas (2.0.2), matplotlib (3.7.1), opencv-python (4.8.0), ChallengeR (1.0.5), and plotly (5.15.0). Biorender was used to create Fig. 1.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The training and validating datasets used in this study are available in the public domain and can be downloaded via the links provided in Supplementary Tables 16 and 17. Source data are provided with this paper in the Source Data file. We confirmed that All the image datasets in this study are publicly accessible and permitted for research purposes. Source data are provided in this paper.

Code availability

The training script, inference script, and trained model have been publicly available at https://github.com/bowang-lab/MedSAM. A permanent version is released on *Zenodo*[47].

References

1. Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J. & Maier-Hein, K. H. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat. Method.* **18**, 203–211 (2021).
2. De Fauw, J. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nat. Med.* **24**, 1342–1350 (2018).
3. Ouyang, D. Video-based AI for beat-to-beat assessment of cardiac function. *Nature* **580**, 252–256 (2020).
4. Wang, G. Deepigeos: a deep interactive geodesic framework for medical image segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**, 1559–1572 (IEEE, 2018).
5. Antonelli, M. The medical segmentation decathlon. *Nat. Commun.* **13**, 4128 (2022).
6. Minaee, S. Image segmentation using deep learning: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 3523–3542 (IEEE, 2021).
7. Kirillov, A. et al. Segment anything. In *IEEE International Conference on Computer Vision*. 4015–4026 (IEEE, 2023).
8. Zou, X. et al. Segment everything everywhere all at once. In *Advances in Neural Information Processing Systems* (MIT Press, 2023).
9. Wang, G. Interactive medical image segmentation using deep learning with image-specific fine tuning. In *IEEE Transactions on Medical Imaging* **37**, 1562–1573 (IEEE, 2018).
10. Zhou, T. Volumetric memory network for interactive medical image segmentation. *Med. Image Anal.* **83**, 102599 (2023).
11. Luo, X. Mideepseg: Minimally interactive segmentation of unseen objects from medical images using deep learning. *Med. Image Anal.* **72**, 102102 (2021).
12. Deng, R. et al. Segment anything model (SAM) for digital pathology: assess zero-shot segmentation on whole slide imaging. Preprint at https://arxiv.org/abs/2304.04155 (2023).
13. Hu, C., Li, X. When SAM meets medical images: an investigation of segment anything model (SAM) on multi-phase liver tumor segmentation. Preprint at https://arxiv.org/abs/2304.08506 (2023).
14. He, S., Bao, R., Li, J., Grant, P.E., Ou, Y. Accuracy of segment-anything model (SAM) in medical image segmentation tasks. Preprint at https://doi.org/10.48550/arXiv.2304.09324 (2023).
15. Roy, S. et al. SAM.MD: zero-shot medical image segmentation capabilities of the segment anything model. Preprint at https://arxiv.org/abs/2304.05396 (2023).
16. Zhou, T., Zhang, Y., Zhou, Y., Wu, Y. & Gong, C. Can SAM segment polyps? Preprint at https://arxiv.org/abs/2304.07583 (2023).
17. Mohapatra, S., Gosai, A., Schlaug, G. Sam vs bet: a comparative study for brain extraction and segmentation of magnetic resonance images using deep learning. Preprint at https://arxiv.org/abs/2304.04738 (2023).
18. Chen, J., Bai, X. Learning to" segment anything" in thermal infrared images through knowledge distillation with a large scale dataset SATIR. Preprint at https://arxiv.org/abs/2304.07969 (2023).
19. Tang, L., Xiao, H., Li, B. Can SAM segment anything? when SAM meets camouflaged object detection. Preprint at https://arxiv.org/abs/2304.04709 (2023).
20. Ji, G.-P. et al. SAM struggles in concealed scenes–empirical study on" segment anything". Science China Information Sciences. **66**, 226101 (2023).
21. Ji, W., Li, J., Bi, Q., Li, W., Cheng, L. Segment anything is not always perfect: an investigation of SAM on different real-world applications. Preprint at https://arxiv.org/abs/2304.05750 (2023).
22. Mazurowski, M. A. Segment anything model for medical image analysis: an experimental study. *Med. Image Anal.* **89**, 102918 (2023).
23. Huang, Y. et al. Segment anything model for medical images? *Med. Image Anal.* **92**, 103061 (2024).
24. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. European Conference on Computer Vision*. 801–818 (IEEE, 2018).
25. Dosovitskiy, A. et al. An image is worth 16x16 words: transformers for image recognition at scale. In: *International Conference on Learning Representations* (OpenReview.net, 2020).
26. Tancik, M. Fourier features let networks learn high frequency functions in low-dimensional domains. In *Advances in Neural Information Processing Systems* **33**, 7537–7547 (Curran Associates, Inc., 2020).
27. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates, Inc., 2017).
28. He, B. Blinded, randomized trial of sonographer versus AI cardiac function assessment. *Nature* **616**, 520–524 (2023).
29. Eisenhauer, E. A. New response evaluation criteria in solid tumours: revised recist guideline (version 1.1). *Eur. J. Cancer* **45**, 228–247 (2009).
30. Ma, J. & Wang, B. Towards foundation models of biological image segmentation. *Nat. Method.* **20**, 953–955 (2023).
31. Ma, J. et al. The multi-modality cell segmentation challenge: towards universal solutions. Preprint at https://arxiv.org/abs/2308.05864 (2023).
32. Xie, R., Pang, K., Bader, G.D., Wang, B. Maester: masked auto-encoder guided segmentation at pixel resolution for accurate, self-supervised subcellular structure recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3292–3301 (IEEE, 2023).
33. Bera, K., Braman, N., Gupta, A., Velcheti, V. & Madabhushi, A. Predicting cancer outcomes with radiomics and artificial intelligence in radiology. *Nat. Rev. Clin. Oncol.* **19**, 132–146 (2022).
34. Clark, K. The cancer imaging archive (TCIA): maintaining and operating a public information repository. *J. Digit. Imaging* **26**, 1045–1057 (2013).
35. Ba, J.L., Kiros, J.R., Hinton, G.E. Layer normalization. Preprint at https://arxiv.org/abs/1607.06450 (2016).
36. He, K. et al. Masked autoencoders are scalable vision learners. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16000–16009 (IEEE, 2022).

37. Loshchilov, I., Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (OpenReview.net, 2019).
38. He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 770–778 (IEEE, 2016).
39. Iakubovskii, P. Segmentation models pytorch. *GitHub* https://github.com/qubvel/segmentation_models.pytorch (2019).
40. Milletari, F., Navab, N., Ahmadi, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *International Conference on 3D Vision* (3DV). 565–571 (IEEE, 2016).
41. Ma, J. Loss odyssey in medical image segmentation. *Med. Image Anal.* **71**, 102035 (2021).
42. Ahmed, A. Radiomic mapping model for prediction of Ki-67 expression in adrenocortical carcinoma. *Clin. Radiol.* **75**, 479–17 (2020).
43. Moawad, A.W. et al. Voxel-level segmentation of pathologically-proven Adrenocortical carcinoma with Ki-67 expression (Adrenal-ACC-Ki67-Seg) [data set]. https://doi.org/10.7937/1FPG-VM46 (2023).
44. Yushkevich, P.A., Gao, Y., Gerig, G. Itk-snap: an interactive tool for semi-automatic segmentation of multi-modality biomedical images. In *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 3342–3345 (IEEE, 2016).
45. Maier-Hein, L. et al. Metrics reloaded: Pitfalls and recommendations for image analysis validation. Preprint at https://arxiv.org/abs/2206.01653 (2022).
46. DeepMind surface-distance. https://github.com/google-deepmind/surface-distance (2018).
47. Ma, J. bowang-lab/MedSAM: v1.0.0. https://doi.org/10.5281/zenodo.10452777 (2023).

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC, RGPIN-2020-06189 and DGECR-2020-00294) and CIFAR AI Chair programs. The authors of this paper highly appreciate all the data owners for providing public medical images to the community. We also thank Meta AI for making the source code of segment anything publicly available to the community. This research was enabled in part by computing resources provided by the Digital Research Alliance of Canada.

Author contributions

Conceived and designed the experiments: J.M. Y.H., C.Y., B.W. Performed the experiments: J.M. Y.H., F.L., L.H., C.Y. Analyzed the data: J.M. Y.H., F.L., L.H., C.Y., B.W. Wrote the paper: J.M. Y.H., F.L., L.H., C.Y., B.W. All authors have read and agreed to the published version of the manuscript.

Competing interests

The authors declare no competing interests

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-024-44824-z.

Correspondence and requests for materials should be addressed to Bo Wang.

Peer review information *Nature Communications* thanks David Ouyang, and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Hindawi
Journal of Healthcare Engineering
Volume 2022, Article ID 4189781, 16 pages
https://doi.org/10.1155/2022/4189781

Review Article

U-Net-Based Medical Image Segmentation

Xiao-Xia Yin,[1,2] **Le Sun**,[3] **Yuhan Fu**,[1] **Ruiliang Lu**,[4] **and Yanchun Zhang**[1]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*
[2]*College of Engineering and Science, Victoria University, Melbourne, VIC 8001, Australia*
[3]*Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing, China*
[4]*Department of Radiology, The First People's Hospital of Foshan, Foshan 528000, China*

Correspondence should be addressed to Xiao-Xia Yin; xiaoxia.yin@gzhu.edu.cn

Received 26 January 2022; Revised 2 March 2022; Accepted 23 March 2022; Published 15 April 2022

Academic Editor: Hangjun Che

Deep learning has been extensively applied to segmentation in medical imaging. U-Net proposed in 2015 shows the advantages of accurate segmentation of small targets and its scalable network architecture. With the increasing requirements for the performance of segmentation in medical imaging in recent years, U-Net has been cited academically more than 2500 times. Many scholars have been constantly developing the U-Net architecture. This paper summarizes the medical image segmentation technologies based on the U-Net structure variants concerning their structure, innovation, efficiency, etc.; reviews and categorizes the related methodology; and introduces the loss functions, evaluation parameters, and modules commonly applied to segmentation in medical imaging, which will provide a good reference for the future research.

1. Introduction

Interpretation of medical images such as CT and MRI requires extensive training and skills because the segmentation of organs and lesions needs to be performed layer by layer. Manual segmentation means a heavy workload to the doctors, which can introduce bias if it involves the subjective opinions of doctors. To analyze complicated images, it usually requires doctors to make a joint diagnosis, which is time consuming. Furthermore, automatic segmentation is a challenging task, and it is still an unsolved problem for most medical applications due to the wide variety connected with image modalities, encoding parameters, and organic variability.

According to [1], medical imaging increased rapidly from 2000 to 2016. As illustrated in Figure 1(a), retrospective cohort study of patterns of medical imaging between 2000 and 2016 was conducted among 16 million to 21 million patients. These patients were enrolled annually in 7 US integrated and mixed-model insurance health care systems and for individuals receiving care in Ontario, Canada. Relative imaging rates by different imaging modality, such as computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound that are used by adults [18–64 years] annually in US and Ontario are also illustrated in Figures 1(b)–1(d), respectively. The imaging rates (per 1000 people) of CT, MRI, and ultrasound use continued to increase among adults, but at lower pace in more recent years. Whether the observed imaging utilization was appropriate or was associated with improved patient outcomes is unknown.

Nowadays, the application of deep learning technology in medical imaging has attracted extensive attention. How to automatically recognize and segment the lesions in medical images has become one of the issues that concern lots of researchers. Ronneberger et al. [2] proposed U-Net at the MICCAI conference in 2015 to tackle this problem, which was a breakthrough of deep learning in segmentation of medical imaging. U-Net is a Fully Convolutional Network (FCN) applied to biomedical image segmentation, which is composed of the encoder, the bottleneck module, and the decoder. The widely used U-Net meets the requirements of medical image segmentation for its U-shaped structure combined with context information, fast training speed, and

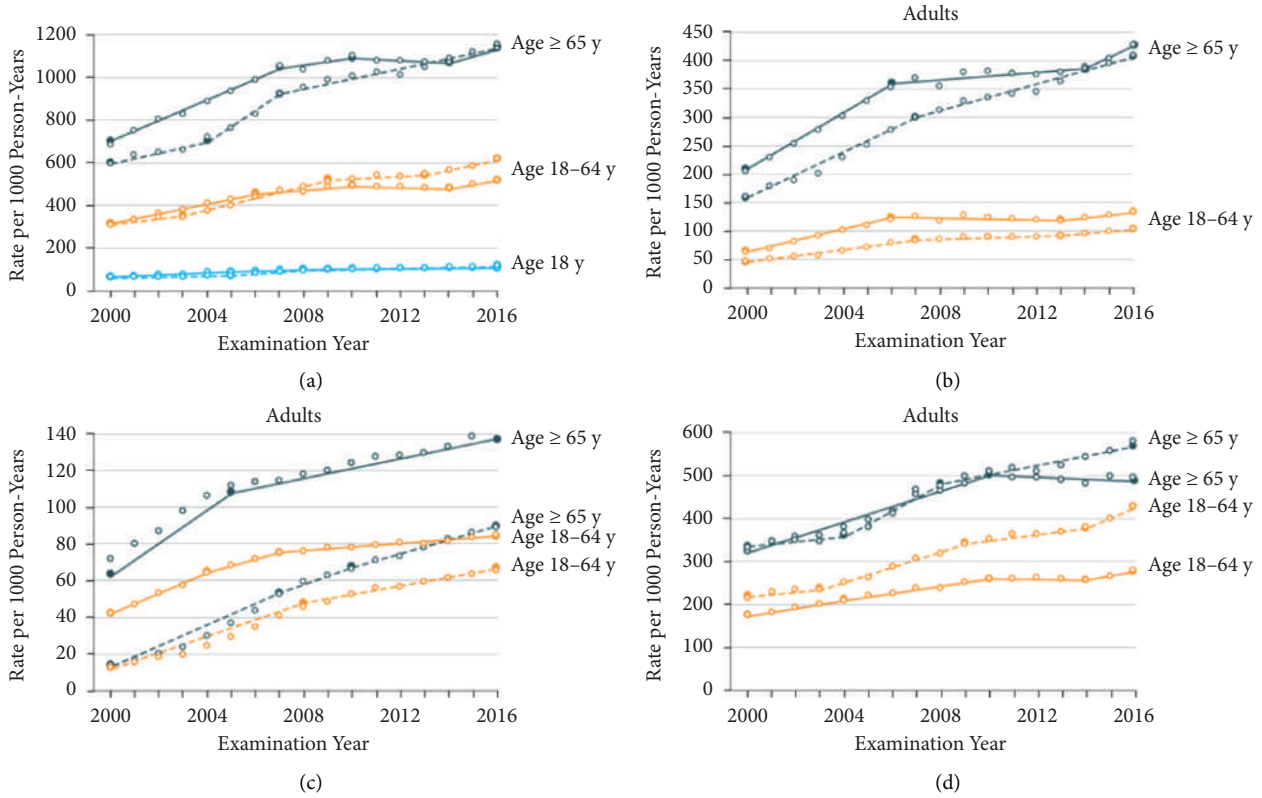


Figure 1: Illustration of relative rates of imaging for United States compared with Ontario from year 2000 to year 2016. CT indicates computed tomography; MRI indicates magnetic resonance imaging. All US data are shown as solid curves; Ontario data are shown as dashed curves [1]. (a) All examinations. (b) CT. (c) MRI. (d) Ultrasound.

a small amount of data used. The structure of U-Net is shown in Figure 2.

Containing many slices, biomedical images are often blocky in a volume space. An image processing algorithm of 2D is often used to analyze a 3D image [3–7]. But when the information is sorted and trained one by one, it would result in increased computational expenses and low efficiency. Therefore, it is difficult to deal with volume images in many cases. A 3D U-Net model derived from the 2D U-Net is designed to address these problems. To further target on architectures of different forms and dimensions, Oktay et al. [8] proposed a new attention gate (AG) model for medical imaging analysis. The model trained with AG indirectly learns to restrain irrelevant regions in an input image and highlight striking features suitable for specific tasks. This is conducive to eradicating the inevitability of applying overt exterior tissue/organ localization units of cascading convolutional neural networks (CNNs) [8, 11]. AG could be combined with standard CNN structure like U-Net, which increases the sensitivity and the precision of the model. To get more advanced data and retain spatial data aimed at 2D segmentation, Gu et al. in 2019 [12] proposed the context encoder network (CE-Net), using pretrained Res-Net blocks as fixed feature extractors. It is mainly composed of three parts—feature encoder, context extractor, and feature decoder. The context extractor is composed of a newly introduced dense atrous convolution (DAC) block and a residual multikernel pooling block (RMP). The introduced CE-Net is widely applied to segmentation in 2D medical imaging [11] and outperforms the original U-Net method.

To further advance the segmentation, UNet++, a novel and greater neural network structure for image segmentation was proposed by Zhou et al. [13]. Moreover, it is a deeply supervised encoder-decoder network connected by a series of nested and dense hopping paths to narrow the semantic gap between the encoding and decoding subnetwork feature maps. Later, to improve more accuracy, especially for organs of different sizes, a new version UNET 3+ was designed by Huang et al. [14]. It utilizes full-scale skip links and deep supervisions, which combines low-level details and high-level semantics mapped at different scales of features and learns hierarchical representation from full-scale aggregated feature maps. The suggested UNet 3+ could increase computational productivity by decreasing network parameters.

Framework regarding nnU-Net ("no-new-Net") is developed by Isensee et al. [15] as a robust self-adaptive framework from U-Net. It was designed by making slight alterations to the 2D and 3D U-Net, where 2D, 3D, 2D, and 3D links were proposed to work together and form a network pool. The nnU-Net could not only automatically adapt its architecture to the given image geometry, but thoroughly define all the other steps including image preprocessing, data training, testing, and potential postprocessing.

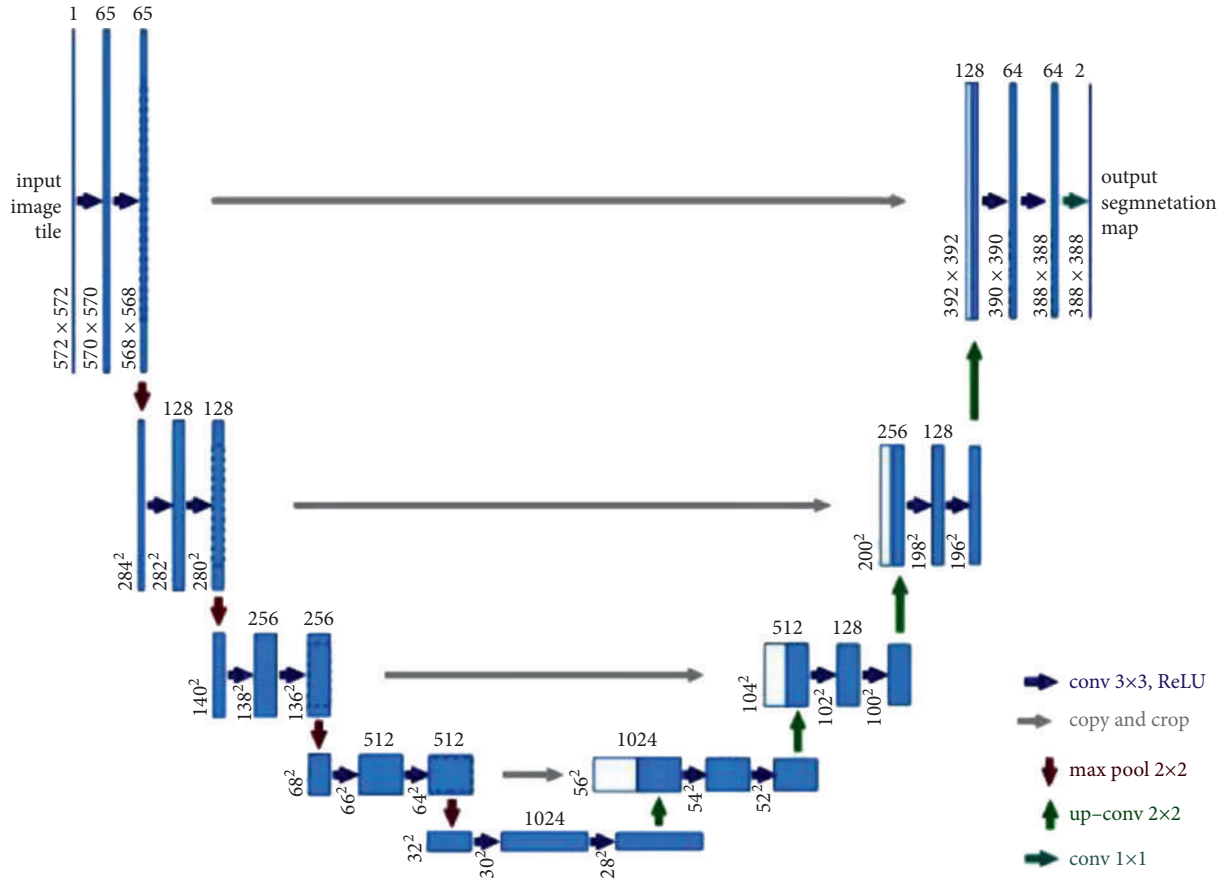


Figure 2: Illustration of U-Net convolution network structure. The left side of the U-shape is the encoding stage, also called contraction path with each layer consisting of two 3 ∗ 3 convolutions with ReLu activation and a 2 ∗ 2 maximum pooling layer. The right side of the U-shape, also called expansion part, consists of the decoding stage and the upsampling process that is realized via 2 ∗ 2 deconvolution to reduce the quantity of input channels by half [2].

U2-Net as a simple and powerful deep network architecture developed by Qin et al. [16] consists of a two-level nested U-shaped structure applied to salient target detection (SOD). It has the following advantages: (1) due to the mixed receptive fields of various sizes in the proposed residual U-shaped block (RSU), it could capture a larger amount of contextual data at various scales. (2) The pooling operation used in the RSU block increases the depth of the entire structure without substantially pushing up the computational cost.

TransUNet designed by Chen et al. [17] encodes tokenized image patches and extracts global contexts from the input sequence of CNN feature map; the decoder upsamples the encoded features and combines with the high-resolution CNN feature maps for precise localization. It uses transformers as a powerful encoding structure for segmentation. Due to the inherent locality of convolution operations, U-Net usually shows limitations in clearly modeling dependencies. The transformer designed for sequence-to-sequence prediction has become an alternative architecture with an innate global self-attention mechanism while localization capabilities of the transformer frame may be limited due to insufficient low-level details.

Since U-Net was proposed, its encoder-decoder-hop network structure has inspired a large amount of segmentation means in medical imaging. Such deeplearning technologies as attention mechanism, dense module, feature enhancement, evaluation function improvement, and other basic U-Net structures have been introduced into medical image segmentation and become widely adopted. These variations of U-Net-related deep learning networks are designed to optimize results by improving the accuracy and computing efficiency of medical image segmentation through changing network structures, adding new modules, etc. However, most of the existing literature related to U-Net focused on introducing isolated new ideas and rarely gave a comprehensive review that summarizes the variations of the U-Net structure for deep learning of segmentation in medical imaging. This paper discussed some of these ideas in more depth.

To sum up, the basic motivation behind this work is not to elaborate into new ideas in U-Net-related deep learning networks but to use effectively U-Net-related deep learning networks techniques into the segmentation of multidimensional data for biomedical applications. The presented method can be generalized to any dimension and can be

used effectively to other types of multidimensional data as well.

This paper is organized as follows. Section 2 addresses the current challenges faced by medical image segmentation. Section 3 reviews these variations of U-Net-related deep learning networks. Section 4 collects various experiment results in literature in relation to different U-Net networks, along with the validation parameters for optimized network structure through the associated deep learning models. The future development in the U-Net-based variant networks is analyzed and discussed. Finally, Section 5 concludes this paper.

2. Existing Challenges

This section presents the current challenges faced by medical image segmentation which make it inevitable to improve and innovate U-Net-based deep learning approaches.

First, medical image processing requires extremely high accuracy for disease diagnosis [18–23]. Segmentation in medical imaging refers to pixel-level or voxel-level segmentation. Generally, the boundary between multiple cells and organs is difficult to be distinguished on the image [3]. Moreover, the data obtained from the image are usually preprocessed, the relevant network is built, which continues to be run by adjusting the parameters even though a certain level of accuracy is reached by using the relevant deep learning model [24].

Second, medical images are acquired from various medical equipment and the standards for them and annotations or performance of CT/MRI machines are not uniform. Hence deep-learning-related trained models are only suitable for specific scenarios. Meanwhile, the deep network with weak generalization may easily capture wrong features from the analyzed medical images. Furthermore, significant inequality always exists between the size of negative and positive samples, which may have a greater impact on the segmentation. However, U-Net could afford an approach achieving better performance in reducing overfitting [25].

Third, interpretable deep learning models applied to analyze medical images are highly required, but there is a lack of confidence in its predicted results [26, 27]. U-Net is a CNN showing poor interpretability. Segmentation in medical imaging could reflect the patient's physiological condition and accurate disease diagnosis. It is not easy for the segmentation lacking interpretability and confidence to be trusted and recognized by professional doctors for clinic application. Although disease diagnosis mainly relies on images, combined with other supplements, which has also increased the complexity. It is a challenge to realize the interpretability and confidence of medical image segmentation via perceiving and adjusting these trade-offs.

3. Methodology

Various medical image segmentation methods have been developed very quickly based on U-Net for performance optimization. U-Net is improved in the areas of application range, feature enhancement, training speed optimization, training accuracy, feature fusion, small sample training set, and generalization improvement. Various strategies are applied in the designing of different network structures to address different segmentation problems.

This section is focused on variations of U-Net-based networks, with the description of U-Net framework, followed by the comprehensive analysis of the U-Net variants by performing (1) intermodality and (2) intramodality categorization to establish better insights into the associated challenges and solutions. The main related work is summarized from the aspects of the improved performance indicators and the main structural characteristics.

3.1. Traditional U-Net. The traditional U-Net is two-dimensional network architecture whose structure is shown in Figure 2. U-Net modifies and extends the Fully Convolutional Network (FCN), making it work with very few training images and produce more accurate segmentation. The major idea is to replace the general shrinkage network with sequential layers and the pooling operation is related to downsampling operator, which is supplemented by upsampling operator. Hence the output's resolution is raised by these layers. The high-resolution of the contracted path is combined with the upsampled output for localization. Hence sequential convolutional layers could study fine features and result in a more accurate segmentation.

An important modification in the U-Net architecture lies in the upsampling section, where there are huge amounts of feature channels allowing the network to spread contextual data to higher-resolution layers. Therefore, the expansion path is roughly symmetrical to the contraction path, forming a U-shaped structure. The network applies the effective part of every convolution—the map of segmentation contains mere pixels, and the complete context of the pixels could be obtained in the input image. This method allows seamless segmentation in arbitrarily large imaging using crucial overlapping tiling strategies, without which the resolution will be limited by GPU memory [1].

The traditional CNN is usually connected to several fully connected layers after convolution and the feature map produced by the convolutional layer is mapped into a feature vector with a fixed length for image-level classification. An improved FCN structure, however, identifies the image at the pixel level, thereby facilitating the task of segmentation in imaging at the semantic level [28].

U-Net could be applied to the segmentation due to its large measurement size of medical images. It is impossible to input the large medical images into the network when they are segmented and required to be cut into small pieces. Overlapping-tilling strategies are suitable for small pieces cutting using U-Net due to its network structure. Thus, it could accept images of any size as inputs [29].

3.2. 3D U-Net. Biomedical imaging is a set of three-dimensional images composed of slices at different locations. Biomedical image analysis involves dealing with a large amount of volume data. Annotating these data labeled by segmentation could cause difficulties because only two-

dimensional slices can be displayed on computers. Therefore, low efficiency and loss of contexts are common during 3D-image processing by traditional 2D image models. To solve this, Ozgun Cicek et al. [30] put forward a 3D U-Net with a shrinking encoder part for analyzing the entire image and a continuous expansion decoder part for generating full-resolution segmentation on the basis of the previous U-Net structure. The structure of 3D U-Net is similar to 2D U-Net in many aspects, except that all operations in the 3D network are replaced with corresponding 3D convolution, 3D pooling, and 3D upsampling. Batch normalization (BN) [31] is used to prevent the network bottlenecks.

Just like the standard U-Net, there is an encoding path and a decoding path with 4 parsing steps in every layer in the encoding path. It contains two $3 \times 3 \times 3$ convolutions followed by a corrected linear unit (ReLu) and then a $2 \times 2 \times 2$ maximum pooling layer with 2-step size of each. Every layer in the synthesis path is composed of $2 \times 2 \times 2$ upper convolutions with two steps in each dimension and two subsequent $3 \times 3 \times 3$ convolutions with a ReLu active layer behind each. The skip connections from the equal-resolution feature map in the encoding path provide the necessary high-resolution features for the decoding path. In the last layer, $1 \times 1 \times 1$ convolution decreases the quantity of output channels to that of labels standing at 3. The structure has 19069955 parameters in total.

In addition to the rotation, scaling, and gray value increase, smooth dense deformation fields are applied to the data and ground truth labelers before training Therefore, random vectors are sampled from a general distribution whose standard deviation is 4 in a grid spaced 32 voxels in each direction, followed by the application of B-spline interpolation. The softmax with weighted cross-entropy loss is used to compare the network output and the ground truth label, to reduce the weight of the common background, increase the weight of internal tubules, and realize the balance effect of small blood vessels and background voxels on the loss.

This end-to-end learning strategy could use semiautomatic and completely automatic methods to segment 3D targets from sparse annotations. The structure and data enhancement of this network allow it to learn from a small number of labeled data and to obtain good generalization capabilities. Appropriate rigid transformation and minor elastic deformation applications could generate reasonable images, rationalize its preprocessing method, and enable the network structure to be extended to any size of the 3D data set.

3.3. Attention U-Net. Attention could be considered as a method of organizing computational resources to interpret the signal informatively. Since its introduction, the attention mechanism has become more and more popular in the deep learning industry. This paper summarizes a method in the application of the attention mechanism onto the U-Net network. Given the small lesions and large shape changes, the attention module is generally added in image segmentation before the encoder- and decoder-related features are stitched or at the bottleneck of U-Net to reduce false-positive predictions.

The Attention U-Net put forward by Oktay et al. [8] in 2018 adds an integrated attention gate (AG) before U-Net splices the corresponding features in the encoder and decoder and readjusted the output features of the encoder. This module facilitates generation of gating signal to eliminate the response of irrelevant and noisy ambiguity in the skip connection, emphasizing the salient features transmitted via the skip connection. Figure 2 displays the inside structure of the attention module.

The salient features useful for specific tasks are stressed in the model trained by AG, which indirectly learns and suppresses unconcerned areas of the input image. Thus, obvious exterior tissue/organ positioning modules are not necessarily used in the Cascaded CNN. Without extra computational cost, the forecast precision and sensitivity of the model could be improved by AG due to its compatibility in standard CNN architectures like U-Net. To estimate the attention U-Net structure, two big CT abdominal data sets were used for multiclass segmentation in imaging. The results show a significant enhancement of U-Net's prediction performance by AG under different data sets and training scales, and the computational efficiency is maintained as well.

The structure of attention U-Net, as shown in Figure 3, is a U-Net-based structure with two stages: encoding and decoding. The coarse-grained map of the left structure captures information in the context and highlights the type and position of foreground objects. Subsequently, feature maps extracted from numerous scales are fused via jump links to merge coarse-grained and fine-grained dense predictions. As for the method put forward in the paper, the attention gate mechanism is to add an AG to each skip connection layer to spread the attention coefficient. AG has two inputs, x from the feature map of the shallow network on the left and g from that of the lower network, which will be output from AG. Then the feature fusion is performed on the feature map after sampling on the right.

This method makes it unnecessary to utilize external object positioning models. It is a convenient tool not only used in natural image analysis and machine translation but also in image classification and regression. Studies showed that the algorithm is very useful for the identification and positioning of tissues/organs, and a certain degree of accuracy could be achieved in the use of smaller computing resources, especially for small-sized organs such as the pancreas [32].

3.4. CE-Net. A fusion of features with different scales serves as a crucial approach to optimizing segmentation performance. Due to fewer convolutions, the low-level features experience lower semantics and more noise despite of their higher resolution and more position. In addition, the resolution is considerably low and the detail perception is poor despite that high-level features contain more intensive semantic information. It is of huge significance to efficiently combine the advantages of these two to improve the

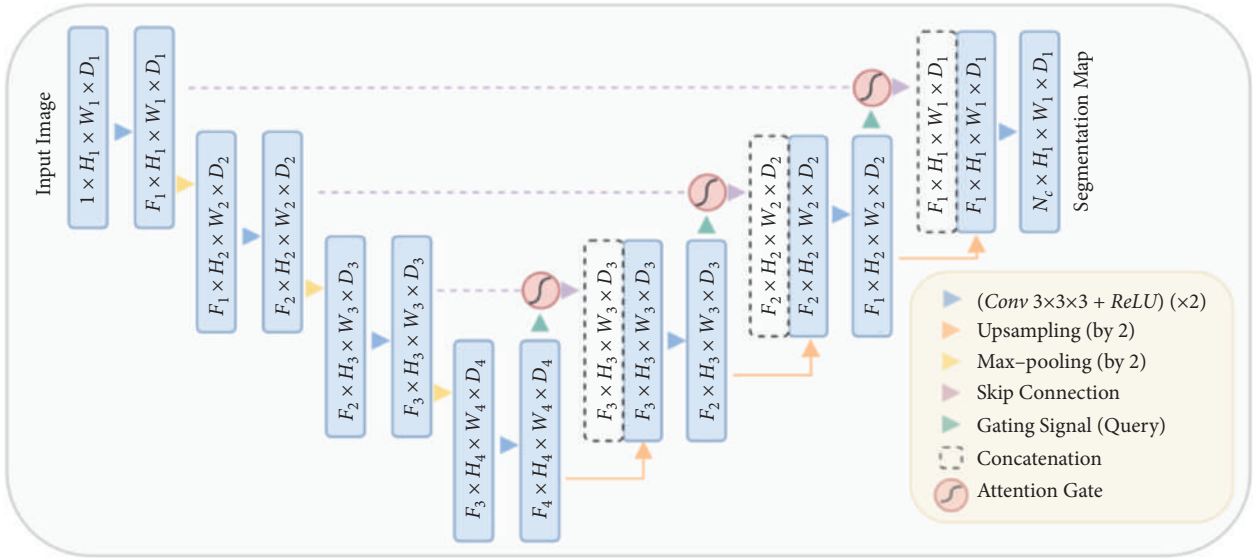


Figure 3: The U-Net model structure of the proposed AG is added. The input image is gradually filtered and downsampled at each scale in the network's encoding part (for example, H4 = H1/8), indicating the quantity of classes. The gates (AGs) filter the characteristics of propagation by skipping connections. The feature AGs is selected by extracting context information (gating) from a coarser scale [8].

segmentation model. Feature fusion includes the contextual features' fusion of the network and the fusion of different modal features in a larger sense. Gu et al. [10] designed a new network called CE-Net, which adopts new modules of dense atrous convolution block (DAC) and residual multikernel pooling block (RMP) to offer fused information like the fusion of contextual features from the encoder, to get higher-level information with a decrease in the feature loss [33], for example, to retain spatial information for 2D segmentation in medical imaging and classification [34].

The overall framework of CE-Net is shown in Figure 4. The DAC block could identify broader and more in-depth semantic features via injecting four cascaded branches with multiscale dense hole convolution. The remaining connections are used to prevent the gradient from disappearing. In addition, the RMP block is a residual multicore pool based on the spatial pyramid pool, which encodes the multiscale context features of the object extracted from the DAC module without extra learning weights using various size pool operations. In summary, the DAC block extracts rich feature representations through multiscale dense hole convolution and then uses the RMP block to extract more context information through multiscale pooling operations. The joint use of newly proposed DAC block and RMP block with the backbone codec structure is unprecedented in CE-Net's context encoder network. This allows the enhancement of the segmentation by further collecting abstract features and maintaining more spatial information.

3.4.1. Feature Encoder Module. In the U-Net structure, each encoder block includes two convolutional layers and a maximum pooling layer. As for the CE-Net network structure, a pretrained ResNet-34 is used in the feature encoding module and the first four feature extraction blocks are retained without mean pooling and full connection. Res-Net adds a shortcut mechanism to avoid gradient disappearance and improve the network convergence efficiency, as shown in Figure 4(b). It is a basic method to improve U-Net segmentation performance using pretrained Res-Net.

3.4.2. Context Extraction Module. The context extraction module, composed of DAC and RMP, extracts contextual semantic information and produces more advanced feature maps.

(1) Hollow Convolution. As for semantic segmentation and object detection, deep convolutional layers have displayed superiority in image feature representation extraction. But the pooling layer might cause loss of image semantic information, which is solved by applying dense hole convolution [35] to dense image segmentation. The hole convolution has an expansion rate parameter which implies that the size of the expansion and the convolution kernel is the same with the ordinary convolution. It means parameters remain unchanged in the neural network, but the hole convolution has a larger receptive field, which refers to the size involved by the convolution kernel on the image. The size of the receptive field is related to stride, the number of convolutional layers, and padding parameters.

(2) DAC. Inspired by Inception [36, 37], Res-Net [38], and hole convolution, dense hole convolution blocks (DAC) [11] are used for encoding high-level semantic feature maps. The DAC has four branches cascading down, with the acceptance field of each branch being 3, 7, 9, and 19, respectively and a gradual increase in the number of atrous convolutions. DAC uses different receptive fields like the inception structure. In each hole convolution branch, a 1×1 convolution is applied to ReLu. The shortcut links in Res-Net are used directly to add the original features. Generally, the convolution of the

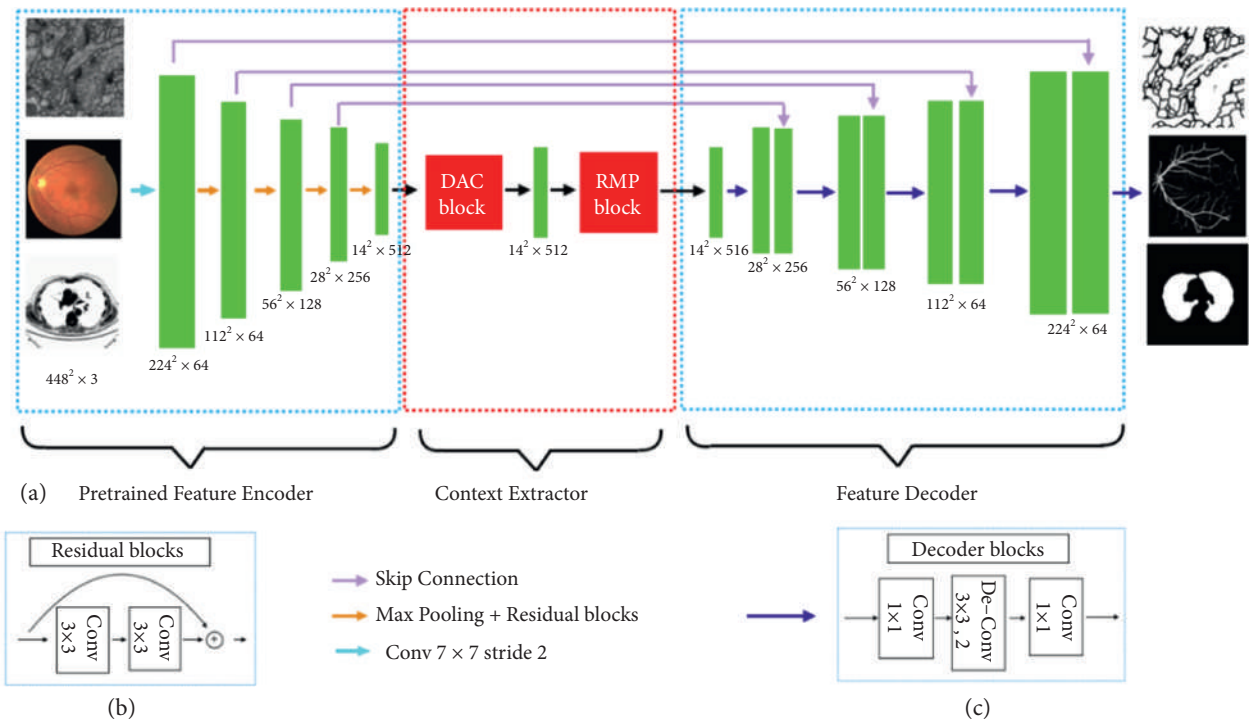


FIGURE 4: CE-net network structure diagram. (a) The original U-Net encoder block is first supplemented by the ResNet-34 block, shown as (b), to be pretrained by ImageNet. A dense convolution (DAC) block and a RMP block were contained in the bottleneck module. Eventually, the features are withdrawn and gathered in the decoder module. The feature size is enlarged by a decoder block (c), including 1×1 convolution and 3×3 deconvolution operations, to supplement the original upsampling operation [11].

large receptive field could extract and produce a larger number of abstract features for the large target and vice versa. The DAC block can extract features from the targets of various sizes through the combination of hole convolutions and different expansion rates.

(3) RMP. One of the challenges in medical image segmentation lies in the significant change in target size [39, 40]. For instance, an advanced tumor is usually much bigger than the early one [41]. An RMP [11] is proposed to solve this problem, by which targets with various sizes could be detected by applying numerous effective fields of view. The proposed RMP utilizes four receptive fields with different size to encode global context information. To reduce the dimensionality of the weights and the computational cost, a 1×1 convolution is used after each pooling branch. Afterwards, the upsampling of the low-dimensional feature map is performed to obtain the same size of features as an original feature map through bilinear interpolation, allowing extraction of features of various scales.

3.4.3. Feature Decoder Module. The feature decoder module allows the recovery of the high-level semantic features extracted from the context extractor module and the feature encoder module. Continuous pooling and convolution operations often lead to the loss of information, which, however, can be remedied by conducting a quick connection from the encoder to the decoder. In U-shaped networks, the two basic operations of decoder are simple upsampling and deconvolution. The image can be enlarged by conducting upsampling through linear interpolation. Deconvolution (also known as transposed convolution) uses convolution to expand the image. Adaptive mapping is used in transposed convolution to recover more comprehensive information. Therefore, transposed convolution is implemented to achieve a higher resolution in the decoder. Based on the shortcut connection and the decoder block, the feature decoder module produces a mask of the same size as the original input.

Unlike U-Net, CE-Net applies a pretrained Res-Net block in the feature encoder. The integration of DAC module, RMP module and Res-Net into the U-Net architecture allows it to retain more spatial information. It was suggested that this approach could optimize segmentation in medical imaging for various tasks of optic disc segmentation [42], retinal blood vessel detection [11], lung segmentation [11], cell contour segmentation [35], and retinal OCT layer segmentation [43]. This approach could be extensively utilized in other 2D medical image segmentation tasks.

3.5. UNET++. Variants of encoder and decoder architectures such as U-Net and FCN are found to be the most advanced image segmentation models [44]. These segmentation networks share a common feature—skip connections that link the depth, semantics, and coarse-grained feature maps from the decoder subnetwork together with the shallow, low-level, and fine-grained feature mapping from the encoder subnetwork. More pinpoint precision is needed

in segmentation of lesions or abnormalities in medical images needs than regular images. Edge segmentation faults in medical imaging may cause some serious consequences in clinic. Therefore, a variety of methods to improve feature fusion have been proposed to address that. In addition, Zhou et al. [13, 45] improved the skip connection and proposed UNet++ with deep monitoring nested dense jump connection path.

As for U-Net, the feature map of the encoder is received by the decoder. But UNet++ uses a dense convolutional block and the quantity of convolutional layers relies on that of the U-shaped structure. In essence, the dense convolution block connects the semantic gap between the encoder and decoder feature maps. It is assumed that when the received encoder feature map and the related decoder feature map are similar at the semantic level, the optimizer can easily tackle the problems it encounters. The effective integration of U-Nets of different depths is used to alleviate unknown network depths. These U-Nets could share an encoder in part and simultaneously learn together through deep supervision, which will allow the model to be pruned and improved. This redesigned skip connection could aggregate semantic features of different scales on the decoder subnet, thereby automatically generating a highly flexible feature fusion scheme.

3.6. UNET 3+. UNet++, an improvement based on U-Net, was designed by developing a structure with nested and dense skip connections. But it does not express enough information from multiple scales and the network parameters are numerous and complex. UNet 3+ (UNet+++) is an innovative network structure proposed by Huang et al. [46], which uses full-scale skip connections and deep supervisions. Full-scale jump connection combines high-level semantics with low-level semantics from feature maps of various scales. Deep supervision learns hierarchical representations from feature maps aggregated at multiple scales. This method uses the newly proposed hybrid loss function to refine the results, particularly suitable for resolving organs of different sizes. It not only improves accuracy and computational efficiency, but also reduces network parameters after fewer channels compared to U-Net and UNet++. The network structure of UNet 3+ is shown in Figure 5.

To learn hierarchical representation from full-scale aggregated feature maps, UNet 3+ further adopts full-scale deep supervision. Different from UNet++, each decoder stage in UNet 3+ has a side output, which uses standard ground truth for supervision. To achieve in-depth supervision, the last layer at each decoder stage is sent to an ordinary 3×3 convolutional layer, followed by a bilinear upsampling and a sigmoid function to enlarge it to full resolution.

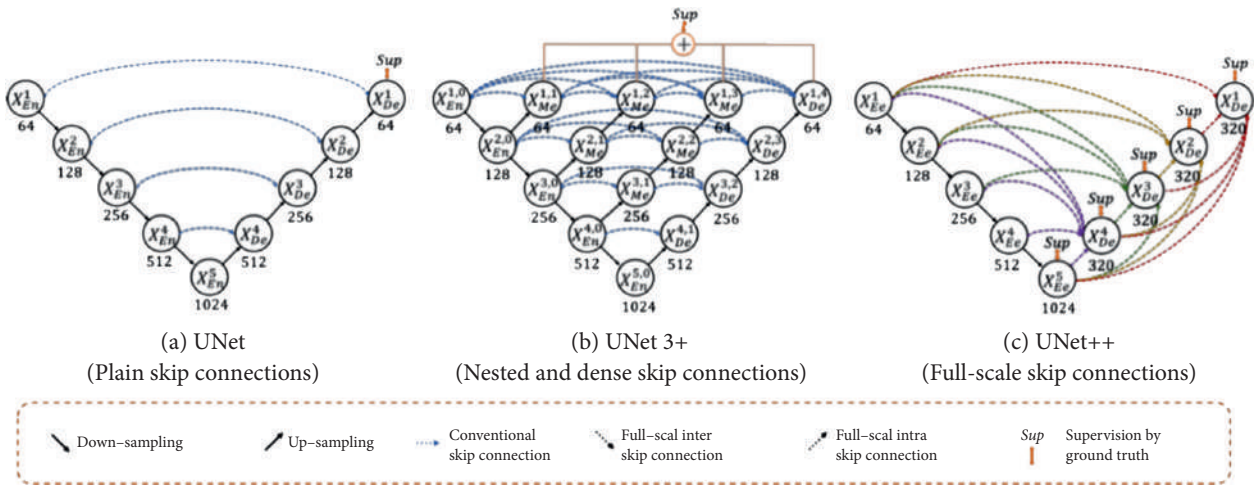
To further strengthen the organ's boundary, a multiscale structural similarity index loss function is proposed to give more weight to the fuzzy boundary. Facilitated by this, UNet 3+ will focus on fuzzy boundaries. The more significant the difference in regional distribution is, the greater the MS-SSIM value becomes [47].

In segmentation of most nonorgan images, false positives are inevitable. The background noise information most likely stays at a shallower level, causing oversegmentation. UNet3++ solves this problem by adding classification-guidance module (CGM) designed to foresee whether the input image has organs to realize more accurate segmentation. With the largest number of semantic information, the classification results could further direct each segmentation side to be output in two steps. With the help of the argmax function, the two-dimensional tensor is converted into a single output of {0, 1}, which represents the presence/absence of organs. Subsequently, the single classification output is multiplied with the side segmentation output. Given the simplicity of the binary classification task, this module could easily obtain accurate classification by optimizing the binary cross-entropy loss function [48] and realize the direction of oversegmentation of nonorgan images.

In summary, UNet 3+ maximizes the application of full-scale feature maps and achieves precise segmentation and efficient network structure with fewer parameters and deep supervision. It has been extensively validated, for example, on representative but demanding volumetric segmentation in medical imaging: (i) liver segmentation from 3D CT scans and (ii) whole heart and big vessels segmentation from 3D MR images [49]. The CGM and the hybrid loss function are further applied to obtain a higher level of accuracy in location-aware and boundary-aware segmented images.

3.7. nnU-Net. It has been designed for different tasks since U-Net was first proposed, with its different network structure, preprocessing, training, and inference. These options are dependent on each other and significant to the final result. Fabian et al. [15, 50] proposed nnU-Net, namely no new-Net. The network is based on 2D and 3D U-Net with a robust self-adaptive framework. It involves a set of three relatively simple U-Net models. Only slight modifications are made to the original U-Net, and no various extension plug-ins were used, including residual connection, dense connection, and various attention mechanisms. The nnU-Net gives unexpectedly accurate results in applications like accurate brain tumor segmentation [51]. Since medical images are often three-dimensional, the design of nnU-Net considers a basic U-Net architecture pool composed of 2D U-Net, 3D U-Net, and U-Net cascade. 2D and 3D U-Net could generate full-resolution results. The first stage of the cascaded network produces a low-resolution result and the second stage optimizes it.

Now that 3D U-Net is widely used, why is 2D still useful? This is because the author proves that when the data are anisotropic, the traditional 3D segmentation method becomes very poor in resolution. The 3D network takes up a lot of GPU memory. Then you could use smaller image slices for training, but for images of larger organs such as livers, this block-based method will hinder training. This is caused by the limited size of the receptive field; the network structure cannot collect enough contextual information to identify the target objects. A cascade model is used here to overcome the shortcomings of 3D U-Net on data sets with large image size.



(a) UNet (Plain skip connections) (b) UNet 3+ (Nested and dense skip connections) (c) UNet++ (Full-scale skip connections)

FIGURE 5: A graphic overview of UNet, UNet++, and UNet 3+. By optimizing jump connections and using full-scale depth monitoring, UNet 3+ integrates multiscale features and produces more precise location perception and segmentation maps with clarified boundaries, regardless of the fewer parameters provided [14, 46].

First, the first-level 3D U-Net is trained on the downsampled image and afterward the result is upsampled to the original voxel interval arrangement. The upsampling result is sent to the second-level 3D U-Net as an additional input channel (one-hot encoding) and the image block-based strategy is used for training on the full-resolution image.

The structure of U-Net has negated most of the new network structures in recent years. It is believed that the network structure has been advanced. The more complex the network, the greater the risk of overfitting. More attention should be paid to other factors such as preprocessing, training, reasoning strategies, and postprocessing.

3.8. U2-Net. Salient object detection (SOD) [52] was designed to segment the most visually attractive objects in the image. It is extensively applied to eye-tracking data [53], image segmentation, and other fields. The recent years have seen a progress in deep CNN especially the emergence of FCN in image segmentation, which substantially enhances the performance of salient target detection. Most SOD network designs share a common pattern, which is to focus on the application of deep features extracted from the present backbone networks, e.g., AlexNet [54, 55], VGG [56], Res-Net [57], ResNeXt [39, 58], and DenseNet [59]. But these backbone networks were proposed for image classification, which extract features that represent semantics instead of local details and global contrast information that are crucial for saliency detection. They must pretrain on the data-inefficient ImageNet data, especially when the target data follows a different distribution from ImageNet.

U2-Net [16, 60] is an uncomplicated and powerful deep network used for salient target detection. It does not use a pretrained backbone model for image classification and could receive training from scratch. It could capture more contextual information because it uses the RSU (ReSidual U-blocks) structure [60, 61], which combines the characteristics of different scales of receptive fields. Meanwhile, it enhances the depth with entire architecture but without significantly increasing computational cost when the pooling operations are applied to these RSU blocks.

RSU structure: as to SOD and other segmentation tasks, both local and global context information is of great significance. As to modern CNN designs, VGG, Res-Net, DenseNet, 1×1 or 3×3 small convolution filters are the most commonly used feature extraction components. Despite its high computational efficiency and small storage size, its filter experience is too small to capture global information; hence, the shallow output feature map only contains local features. To obtain more global information on the shallow high-resolution topographic map [62, 63], the most direct method is to expand the receiving field.

The existing convolutional block with the smallest receptive field fails to obtain global information, and the output feature map at the bottom layer only contains local features. To obtain richer global information on high-resolution shallow feature maps, the receptive field must be expanded. There are attempts to expand the receptive field by using hole convolution to extract local and nonlocal features. However, performing multiple extended convolutions on the input feature map of the original resolution (especially in the initial stage) requires a large amount of computing and memory resources. Inspired by U-Net, a new RSU is proposed to obtain multiscale features within the stage. RSU is mainly composed of three parts as follows.

(1) Input convolutional layer: convert the input feature map $\mathrm{x}(H \times W \times C_{\mathrm{in}})$ into an intermediate image $F_1(x)$ with the number of C_{out} channels to extract local features.

(2) Use the intermediate feature map $F_1(x)$ as input and learn to extract and encode multiscale context information $U(F_1(x))$. U refers to U-Net. The greater the L, the deeper the RSU and the more pooling operations, the bigger the receptive field and the more local and global features.

(3) Through the summation of $F_1(x)$, local features and multiscale features are merged.

Hence the residual U-block RSU about how to stack and connect these structures is proposed. It results in a completely different method from previous cascade stacking: Un-Net. The exponential notation here means a nested U-shaped structure rather than a cascaded stack. In theory, the index n could be adjusted to any positive integer to realize a single-layer or multilayer nested U-shaped structure. However, to be applied to practical applications. n is set to 2 to form the two-leveled U2-Net. The top layer of it is a large U-shaped structure including 11 stages with each filled with a well-configured RSU. Therefore, the nested U structure could extract the multiscale features in each stage and the multilevel features in the aggregation stage with higher efficiency. Unlike those SOD models which are built on present backbones, U2-Net is constructed on the proposed RSU block that allows training from scratch and different model sizes to be configured according to the constraints of the target environment.

3.9. TransUNet. Due to the inherent locality of convolution operations, U-Net is usually limited in explicitly modeling remote dependencies. Recently, the transformer designed for sequence-to-sequence prediction has emerged as an alternative architecture with a global self-attention mechanism. However, its positioning capabilities are limited by its insufficient underlying details. TransUNet with the advantages of transformer [64] and U-Net was proposed by Chen et al. [17] as a powerful alternative to medical image segmentation. This is because the transformer treats the input as a one-dimensional sequence and only focuses on modeling the global context of all stages, which results in low-resolution features and a lack of detailed positioning information. Direct upsampling to full resolution cannot effectively recover this information, which results in rough segmentation results. In addition, the U-Net architecture provides a way to achieve precise positioning by extracting low-level features and linking them to high-resolution CNN feature maps, which could adequately complement for fine spatial details. An overview of the framework is shown in Figure 6.

The transformer could be used as a powerful encoder for medical image segmentation and combined with U-Net to enhance finer details and restore local spatial information. TransUNet has achieved excellent performance in multiorgan segmentation and heart segmentation. In the design of TransUNet, the issue is how to encode the feature representation directly from the decomposed image patch using the transformer.

In order to complete the purpose of segmentation, that is, to classify the image at the pixel level, the most direct method is to upsample the encoded feature map to predict the full resolution of the dense output. To restore the spatial order, the size of the coding function should first reshape the size of the image from HW/P^2 to $H/P \times W/P$. The next step is to use 1×1 convolution to decrease the channel size of the reshaped feature to the number of classes. Afterward, directly upsampling the feature map to full resolution $H \times W$ is performed to predict the final segmentation result.

In summary, TransUNet mixes CNN and transformer as an encoder and allows the use of medium and high-resolution CNN feature maps in the decoding path, hence more context information can be involved. TransUNet not only uses image features as a sequence to encode strong global context but also makes good use of low-level CNN features through a U-shaped hybrid frame design.

4. Overview of Validation Methods of Resultant Experiments

4.1. Evaluation Parameters. The several U-Net-based extended structure networks introduced above possess different improved structures and characteristics, and their effects in real-world applications vary. Therefore, this paper summarized the corresponding advantages of each by comparing the parameters. The segmentation evaluation parameters play a crucial part in the evaluation of image segmentation performance. This section mainly lists several commonly used evaluation parameters in image segmentation neural networks and illustrates the characteristics of each network in various experiments.

True positive (TP), true negative (TN), false positive (FP), and false negative (FN) are mainly used to count two types of classification problems. There is no doubt that multiple categories could also be counted separately. The samples are divided into positive and negative samples.

4.2. Performance Comparison. The related methods proposed in this paper use almost different data sets including retinal blood vessels, liver, kidney, gastric cancer, and cell sections. The data sets used by various methods are not the same; hence, it is difficult to compare different methods horizontally. This paper listed the data sets to provide an index of data set names. The performance comparison is listed in Table 1.

4.3. Future Development. Medical image segmentation is a popular and developing research field. As an implementation standard of medical segmentation, the U-Net network structure has been in use and improved for many years. Although the work and improvements of U-Net in recent years have begun to solve the challenges presented in Section 2, there are still some unsolved problems. In this part, some promising research discussing those problems will be outlined (accuracy issues, interpretability, and network training issues) and other challenges that may still exist will be introduced.

4.3.1. Higher Generalization Ability. The model is not only required to have a good fit (training error) to the training data set but also to have a good fit (generalization ability) to the unknown data set (prediction set). As for tasks like medical image segmentation, small sample data are usually more prone to overfitting or underfitting. Therefore, the frequently used methods such as early stopping, regularization, feedback, input fuzzification, and dropout have

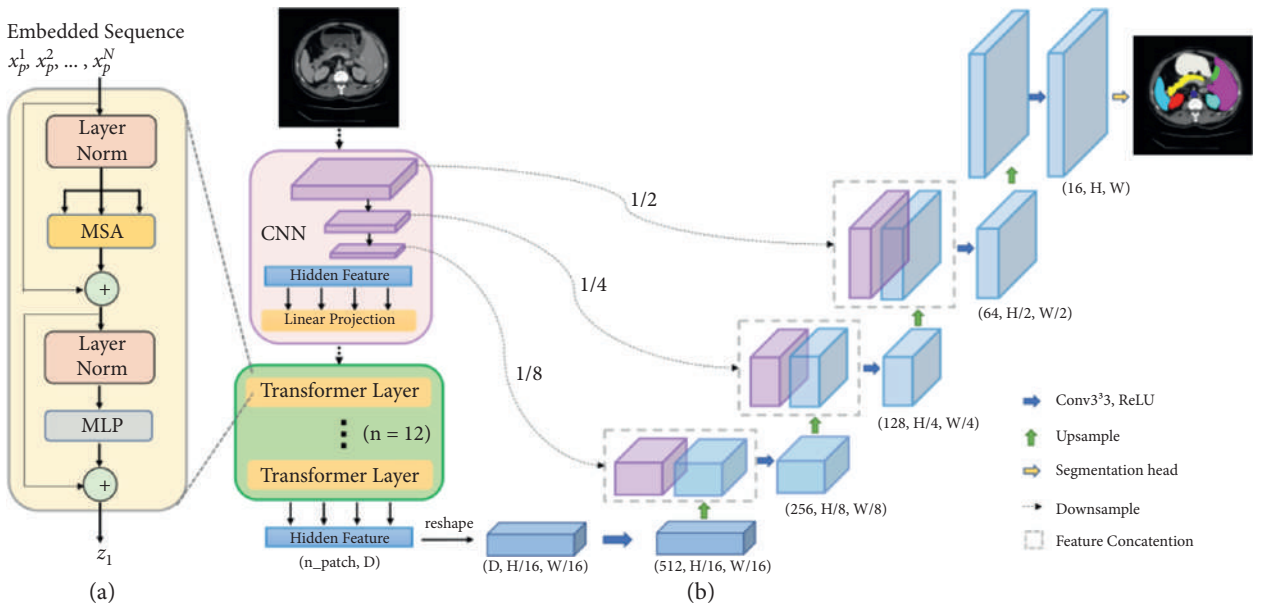


Figure 6: Overview of TransUNet's framework. (a) The transformer layer's structure and (b) the entire TransUNet's structure. After the U-Net encoding stage of the network, a transformer structure composed of 12 layers of transformers is added to process the corresponding processed image sequence. Then the number of channels and dimensions of the picture are unified to the standard by redetermining the size [17].

Table 1: Performance contrast of the networks listed in this article. Different methods use different data sets for evaluation, which makes it hard to compare various approaches horizontally.

U-net type	Medical image data base	Evaluation parameters	Values
U-Net [1]	DRIVE [1] Amazon data set	Accuracy IoU	0.955 ± 0.003 [1] 0.9530 [64]
3D U-Net [29]	Xenopus kidney embryos	IoU	0.732 [29]
Attention U-Net [7]	Gastric cancer [7] Amazon data set [64]	Dice coefficient IoU	0.767 ± 0.132 [7] 0.9581 [64]
CE-Net [10]	DRIVE [10] Lung segmentation CT	Accuracy IoU	0.975 ± 0.003 [10] 0.9495 [65]
U-Net++ [12]	Cell nuclei [12] Lung segmentation CT [65]	Jaccard/IoU IoU	0.9263 [12] 0.9521 [65]
UNET 3+ [13]	ISBI LiTS 2017	Dice coefficient	0.9552
nnU-Net [14]	BRATS challenge	Dice coefficient	0.8987 ± 0.157
U2 Net [15]	Vienna reading [15] CVC-ClinicDB	Dice coefficient IoU	0.8943 ± 0.04 [15] 0.8611 [66]
TransUNet [16]	MICCAI 2015 CVC-ClinicDB	Dice coefficient IoU	0.7748 0.89 [66]

improved the generalization problem of neural networks to varying degrees. But in general, the essence of the neural network is instance learning and the network has the cognition of most instances through limited samples. However, recently it has been suggested to seek innovation and abandon the long-used input vector fuzzification processing method.

4.3.2. Improved Interpretability. As for Interpretability or Explainable Artificial Intelligence (XAI), what always concerns researchers engaged in machine learning is that many current deep neural networks cannot fully understand the decision-making models from human's perspective. We do not know when there will be an error and what causes it in medical images. Medical images reflect on people's health; hence, interpretability is crucial. Now, people often use sensitivity analysis or gradient-based analysis methods for interpretability analysis. There are many attempts to implement interpretability after training such as surrogate models, knowledge distillation, and hidden layer visualization.

4.3.3. Resolution and Processing of Data Imbalance. Data imbalance often occurs due to inconsistent machine models in medical image segmentation. But in fact, many common imbalance problems can be avoided. Nowadays, the

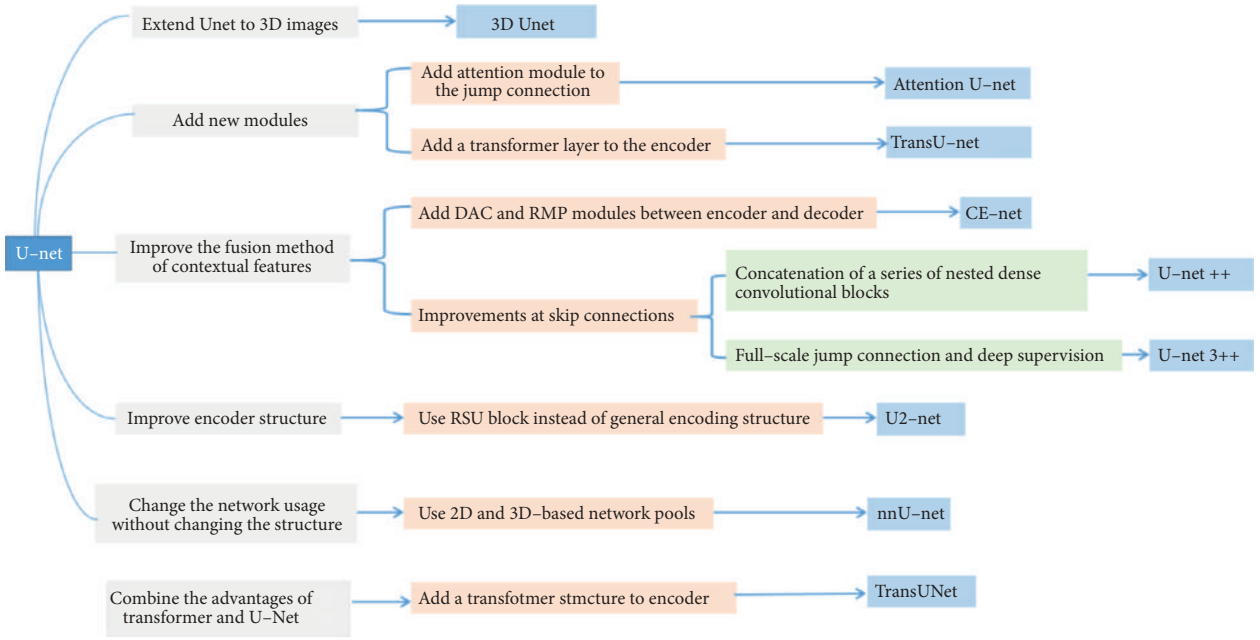


Figure 7: U-Net-based extension structure summary diagram.

Table 2: The summary of the changes in network structures and adjusted parameters. The number of parameters for a $K \times K(\times K)$ size convolution kernel, C_i input channels, and C_o output channels is a $K \times K(\times K) \times C_i \times C_o$ and is given below for a few U-Net variants.

Model structure	Dimension	Improved structure	Highlights	#Params	Kernel size
U-Net	2D	Fully connected layer (relative to CNN)	Fully connected layer changed to upsampling layer	30M [67]	3×3; 2×2; 1×1
3D U-Net	3D	Encoder, decoder	2D convolution operation replaced with 3D	19M [68]	$1 \times 1 \times 1$; $2 \times 2 \times 2$; $3 \times 3 \times 3$
Attention U-Net	2D	Skip connection	Add the attention module to the skip connection	123M [65]	1×1
CE-Net	2D	Bottleneck between encoder and decoder	DAC and RMP structure	110 [65]	3×3; 1×1
UNET++	2D	Skip connection	Use dense blocks and in-depth supervision	35 [65]	3×3; 1×1
UNET 3+	2D	Skip connection	Full-scale jump connection and deep supervision	26.97 [69]	3×3; $3 \times 3 \times 3$
nnU-Net	2D/3d	Network organization	Multiple ordinary U-Nets form a network pool		$4 \times 4 \times 4$
U2-Net	2D	Encoder and decoder	Use RSU as the decoding and encoding unit	176M [70]	3×3
Trans-U-Net	2D	Encoder	Add the transformer module after the decoder	2.93M [66, 71]	1×1

common ways to solve them include expanding the data, using different evaluation indicators, resampling the data set, trying artificial data samples, and using different algorithms. It was suggested in a recent ICML paper that the increased amount of data could increase the error of the training set with a known distribution and destroys the original training set's allocation, thereby improving the classifier's performance. This paper implicitly used mathematical methods to increase the data without changing the size of the data set. However, we believe that destroying the original distribution is beneficial for dealing with imbalances.

4.3.4. A New Exploration of Transformer and Attention Mechanism. This paper introduced attention and transformer methods that afford an innovative combination of these two mechanisms and U-Net. So far, some research has explored the feasibility of using the Transformer structure which only works on the self-attention mechanism as an

encoder for medical image segmentation without any pretraining. In the future, more novel models will be proposed to solve different problems in medical segmentation with continuous breakthroughs in attention and transformer methods.

4.3.5. Multimodal Learning and Application. Single-modal representation learning is to express information as numerical vectors that could be processed by the computer or further abstracted into higher-level feature vectors, while multimodal representation learning is to eliminate intermodality by taking advantage of the complementarity between multiple modalities. In medical images, multimodal data with different imaging mechanisms could provide information at multiple levels. Multimodal image segmentation is used to fuse information among different modalities for multimodal fusion and collaborative learning. Research on multimodal learning is becoming more popular in recent years and the application of medical images will grow more sophisticated in the future.

5. Discussion and Conclusion

This paper introduces several classic networks with improved U-Net structures to deal with different problems that are encountered in medical image segmentation. We review the paper.

A summary of the technical context based on the U-Net extended structure introduced above is shown in Figure 7.

This paper summarized U-Net network dimensions, improved structure, and structure parameters, along with kernel size. Table 2 summarized these aspects.

U- Net could meet the high-precision segmentation of all lesions with its differentiation of organ structures and the diversification of lesion shapes. With the development and improvement of attention mechanism, dense module, transformer module, residual structure, graph cut, and other modules, different modules based on U-Net have been used recently to achieve precise segmentation of different lesions. Based on the various U-Net extended structures, this paper classifies and analyzes several classic medical image segmentation methods based on the U-Net structure.

It is concluded that U-Net-based architecture is indeed quite ground-breaking and valuable in medical image analysis. However, although U-Net-based deep learning has become a dominant method in a variety of complex tasks such as medical image segmentation and classification, it is not all-powerful. It is essential to be familiar with key concepts and advantages of U-Net variants as well as limitations of it, in order to leverage it in radiology research with the goal of improving radiologist performance and, eventually, patient care. Despite the many challenges remaining in deep learning-based image analysis, U-Net is expected to be one of the major paths forward [72–80].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by Science and Technology Projects in Guangzhou, China (grant no. 202102010472). This work is funded by National Natural Science Foundation of China (NSFC) (grant no. 62176071).

References

[1] R. Smith-Bindman, M. L. Kwan, E. C. Marlow et al., "Trends in use of medical imaging in US health care systems and in Ontario, Canada, 2000-2016," *JAMA*, vol. 322, no. 9, pp. 843–856, 2019 Sep 3.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds., October 2015.

[3] X. X. Yin, B. W.-H. Ng, and Q. Yang, A. Pitman, K. Ramamohanarao, and D. Abbott, Anatomical landmark localization in breast dynamic contrast-enhanced MR imaging," *Medical, & Biological Engineering & Computing*, vol. 50, no. 1, pp. 91–101, 2012.

[4] X.-X. Yin, S. Hadjiloucas, J.-H. Chen, Y. Zhang, J.-L. Wu, and M.-Y. Su, "Correction: tensor based multichannel reconstruction for breast tumours identification from DCE-MRIs," *PLoS One*, vol. 12, no. 4, p. e0176133, 2017.

[5] P. Radiuk, "Applying 3D U-net architecture to the task of multi-organ segmentation in computed tomography," *Applied Computer Systems*, vol. 25, no. 1, pp. 43–50, 2020.

[6] Q. Tong, M. Ning, W. Si, X. Liao, and J. Qin, "3D deeply-supervised U-net based whole heart segmentation," in *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges. STACOM 2017*, M. Pop, Ed., vol. 10663, Cham. Switzerland, Springer, 2018.

[7] C. Wang, T. MacGillivray, G. Macnaught, G. Yang, and D. Newby, "A two-stage U-net model for 3D multi-class segmentation on full-resolution cardiac data," in *Statistical Atlases and Computational Models of the Heart. Atrial Segmentation and LV Quantification Challenges. STACOM 2018*, M. Pop, Ed., vol. 11395, Springer, Cham. Switzerland, 2019.

[8] O. Oktay, J. Schlemper, L. Folgoc et al., "Attention U-Net: Learning where to Look for the Pancreas," in *Proceedings of the 1st Conference on Medical Imaging with Deep Learning*, Amsterdam, The Netherlands, July 2018.

[9] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.

[11] P. Jaccard, "The distribution of the flora in the alpine Zone.1," *New Phytologist*, vol. 11, no. 2, pp. 37–50, February 1912.

[12] Z. Gu, J. Cheng, H. Fu et al., "CE-net: context encoder network for 2D medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 38, no. 10, pp. 2281–2292, 2019.

[13] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: a nested U-net architecture for medical image segmentation," *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, vol. 11045, pp. 3–11, 2018.

[14] H. Huang, L. Lin, R. Tong et al., "UNet 3+: a full-scale connected UNet for medical image segmentation," in *Proceedings of the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1055–1059, Barcelona, Spain, May 2020.

[15] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.

[16] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-Net: going deeper with nested U-structure for salient object detection," *Pattern Recognition*, vol. 106, p. 107404, 2020.

[17] J. Chen, Y. Lu, Q. Yu et al., "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," 2021, https://arxiv.org/abs/2102.04306.

[18] X. X. Yin, S. Hadjiloucas, and Y. Zhang, *Pattern Classification of Medical Images: Computer Aided Diagnosis*, Springer-Verlag, Heidelberg, Germany, 2017.

[19] S. Irshad, X. Yin, and Y. Zhang, "A new approach for retinal vessel differentiation using binary particle swarm optimization," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 9, no. 5, pp. 510–522, 2021.

[20] X. Yin, S. Irshad, and Y. Zhang, "Classifiers fusion for improved vessel recognition with application in quantification of generalized arteriolar narrowing," *Journal of Innovative Optical Health Sciences*, vol. 13, no. 01, p. 1950021, 2020.

[21] X. X. Yin, L. Yin, and S. Hadjiloucas, "Pattern classification approaches for breast cancer identification via MRI: state-of-the-art and vision for the future," *Applied Sciences*, vol. 10, no. 20, p. 7201, 2020.

[22] D. Pandey, X. Yin, H. Wang, and Y. Zhang, "Accurate vessel segmentation using maximum entropy incorporating line detection and phase-preserving denoising," *Computer Vision and Image Understanding*, vol. 155, pp. 162–172, 2017.

[23] X. X. Yin, S. Hadjiloucas, Y. Zhang et al., "Pattern identification of biomedical images with time series: contrasting THz pulse imaging with DCE-MRIs," *Artificial Intelligence in Medicine*, vol. 67, pp. 1–23, 2016.

[24] T. J. Sejnowski, "The unreasonable effectiveness of deep learning in artificial intelligence," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30033–30038, 2020.

[25] P. J. R Prasad, O. J Elle, F. Lindseth, F. Albregtsen, and R. P. Kumar, "Modifying U-Net for small data set: a simplified U-Net version for liver parenchyma segmentation," in *Proceedings of the SPIE 11597, Medical Imaging 2021: Computer-Aided Diagnosis*, February 2021.

[26] D. Chen, S. Liu, P. Kingsbury et al., "Deep learning and alternative learning strategies for retrospective real-world clinical data," *Npj Digital Medicine*, vol. 2, no. 1, p. 43, 2019.

[27] M. Reyes, R. Meier, S. Pereira et al., "On the interpretability of artificial intelligence in radiology: challenges and opportunities," *Radiology: Artificial Intelligence*, vol. 2, no. 3, p. e190043, 2020.

[28] S. Zheng, X. Lin, W. Zhang et al., "MDCC-Net: multiscale double-channel convolution U-Net framework for colorectal tumor segmentation," *Computers in Biology and Medicine*, vol. 130, p. 104183, 2021.

[29] X. Liu, L. Song, S. Liu, and Y. Zhang, "A review of deep-learning-based medical image segmentation methods," *Sustainability*, vol. 13, no. 3, p. 1224, 2021.

[30] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-net: learning dense volumetric segmentation from sparse annotation," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, S. Ourselin, L. Joskowicz, M. Sabuncu, G. Unal, and W. Wells, Eds., October, 2016.

[31] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift, ICML'15," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, pp. 448–456, Lille, France, July, 2015.

[32] J. Schlemper, O. Oktay, M. Schaap et al., "Attention gated networks: learning to leverage salient regions in medical images," *Medical Image Analysis*, vol. 53, pp. 197–207, 2019.

[33] H. Ma, Y. Zou, and P. X. Liu, "MHSU-Net: a more versatile neural network for medical image segmentation," *Computer Methods and Programs in Biomedicine*, vol. 208, p. 106230, 2021.

[34] B. Jin, P. Liu, P. Wang, L. Shi, and J. Zhao, "Optic disc segmentation using attention-based U-net and the improved cross-entropy convolutional neural network," *Entropy*, vol. 22, no. 8, p. 844, 2020.

[35] C. Han, Y. Duan, X. Tao, and J. Lu, "Dense convolutional networks for semantic segmentation," *IEEE Access*, vol. 7, pp. 43369–43382, 2019.

[36] R. F. Mansour and N. O. Aljehane, *An Optimal Segmentation with Deep Learning Based Inception Network Model for Intracranial Hemorrhage Diagnosis*, Neural Comput & Applic, London, UK, 2021.

[37] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-first AAAI conference on artificial intelligence*, vol. 4, p. 12, San Francisco, California, USA, February, 2017.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, Las Vegas, Nevada, July, 2016.

[39] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, "Deep learning techniques for medical image segmentation: achievements and challenges," *Journal of Digital Imaging*, vol. 32, no. 4, pp. 582–596, 2019.

[40] T. Zhou, R. Su, and S. Canu, "A review: deep learning for medical image segmentation using multi-modality fusion," *Array*, vol. 3–4, p. 100004, 2016.

[41] T. J. Anchordoquy, Y. Barenholz, D. Boraschi et al., "Mechanisms and barriers in cancer nanomedicine: addressing challenges, looking for solutions," *ACS Nano*, vol. 11, no. 1, pp. 12–18, 2017.

[42] J. Jin, H. Zhu, J. Zhang et al., "Multiple U-Net-Based automatic segmentations and radiomics feature stability on ultrasound images for patients with ovarian cancer," *Frontiers in Oncology*, vol. 10, p. 614201, 2021.

[43] Y. Ma, H. Hao, J. Xie et al., "ROSE: a retinal OCT-angiography vessel segmentation data set and new model," *IEEE Transactions on Medical Imaging*, vol. 40, no. 3, pp. 928–939, 2020.

[44] P. Saiviroonporn, K. Rodbangyang, T. Tongdee et al., "Cardiothoracic ratio measurement using artificial intelligence: observer and method validation studies," *BMC Medical Imaging*, vol. 21, pp. 1–11, 2021.

[45] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, "Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7340–7351, Honolulu, HI, USA, July, 2017.

[46] H. Huang, L. Lin, R. Tong et al., "UNet 3+: a full-scale connected UNet for medical image segmentation," 2020, https://arxiv.org/abs/2004.08790.

[47] G. Mattyus, W. Luo, and R. Urtasun, "DeepRoadMapper: extracting road topology from aerial images," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3438–3446, Venice, Italy, October, 2017.

[48] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.

[49] Q. Dou, L. Yu, H. Chen et al., "3D deeply supervised network for automated segmentation of volumetric medical images," *Medical Image Analysis*, vol. 41, pp. 40–54, 2017.

[50] F. Isensee, R. Sparks, and S. Ourselin, "Batchgenerators — a Python Framework for Data Augmentation," 2020, https://zenodo.org/record/3632567#.YkGUnOdBzIU.

[51] Y. Zhang, S. Liu, C. Li, and J. Wang, "Rethinking the dice loss for deep learning lesion segmentation in medical images," *Journal of Shanghai Jiaotong University*, vol. 26, no. 1, pp. 93–102, 2021.

[52] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: a benchmark," in *Proceedings of the Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., October, 2012.

[53] F. Xiao, L. Peng, L. Fu, and X. Gao, "Salient object detection based on eye tracking data," *Signal Processing*, vol. 144, pp. 392–397, 2018.

[54] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[56] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, vol. 5, pp. 730–734, Kuala Lumpur, Malaysia, November, 2015.

[57] Q. Chen, H. Yue, X. Pang et al., "Mr-ResNeXt: a multi-resolution network architecture for detection of obstructive sleep apnea," in *Neural Computing for Advanced Applications. NCAA 2020. Communications in Computer and Information Science*, H. Zhang, Z. Zhang, Z. Wu, and T. Hao, Eds., vol. 1265, Singapore, Springer, 2020.

[58] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, Honolulu, HI, USA, July, 2017.

[59] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Honolulu, HI, USA, July, 2017.

[60] J. I. Orlando, P. Seebock, H. Bogunovic et al., "U2-Net: a bayesian U-net model with epistemic uncertainty feedback for photoreceptor layer segmentation in pathological OCT scans," in *Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 1441–1445, Venice, Italy, April 2019.

[61] D. Li, D. A. Dharmawan, B. P. Ng, and S. Rahardja, "Residual U-net for retinal vessel segmentation," in *Proceedings of the 2019 IEEE International Conference on Image Processing*, pp. 1425–1429, ICIP), Taipei, Taiwan, September 2019.

[62] A. J. Kent and A. Hopfstock, "Topographic mapping: past, present and future," *The Cartographic Journal*, vol. 55, no. 4, pp. 305–308, 2018.

[63] A. Kent, "Topographic maps: methodological approaches for analyzing cartographic style," *Journal of Map & Geography Libraries*, vol. 5, no. 2, pp. 131–156, 2009.

[64] D. John and C. Zhang, "An attention-based U-Net for detecting deforestation within satellite sensor imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 107, p. 102685, 2022.

[65] R. Su, D. Zhang, J. Liu, and C. Cheng, "MSU-net: multi-scale U-net for 2D medical image segmentation," *Frontiers in Genetics*, vol. 12, p. 639930, 2021.

[66] A.-J. Lin, B. Chen, J. Xu, Z. Zhang, and G. Lu, "DS-TransUNet: Dual Swin Transformer U-Net for Medical Image Segmentation," 2021, https://arxiv.org/abs/2106.06716.

[67] N. Beheshti and L. Johnsson, "Squeeze U-net: a memory and energy efficient image segmentation network," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1495–1504, Seattle, WA, USA, June, 2020.

[68] C. Ozgun, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-net: learning dense volumetric segmentation from sparse annotation," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2016*, pp. 424–432, Springer International Publishing, Athens, Greece, October, 2016.

[69] H. Huang, L. Lin, R. Tong et al., "UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, pp. 1055–1059, Barcelona, Spain, May, 2020.

[70] C. Wang, C. Li, J. Liu et al., "U2-ONet: a two-level nested octave U-structure network with a multi-scale Attention mechanism for moving object segmentation," *Remote Sensing*, vol. 13, no. 1, 2021.

[71] Y. Yang and S. Mehrkanoon, "AA-TransUNet: Attention Augmented TransUNet For Nowcasting Tasks," 2022, https://arxiv.org/abs/2202.04996.

[72] X. Jiang, Y. Wang, Y. Wang, W. Liu, and S. Li, "CapsNet, CNN, FCN: comparative performance evaluation for image classification," *International Journal of Machine Learning and Computing*, vol. 9, no. 6, pp. 840–848, 2019.

[73] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," *and Computers*, vol. 2, pp. 1398–1402, 2003.

[74] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 02, pp. 318–327, 2020.

[75] F. Isensee, P. F. Jäger, P. M. Full, P. Vollmuth, and K. H. Maier-Hein, "nnU-net for brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2020*, A. Crimi and S. Bakas, Eds., vol. 12659, Cham. Switzerland, Springer, 2021.

[76] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale," 2021, https://arxiv.org/abs/2010.11929.

[77] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?," 2019, https://arxiv.org/abs/1905.10650.
[78] J. B. Cordonnier, A. Loukas, and M. Jaggi, "Multi-head attention: collaborate instead of concatenate," 2020, https://arxiv.org/abs/2006.16362.
[79] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, USA, 2009.
[80] L. Liu, J. Cheng, Q. Quan, F.-X. Wu, Y.-P. Wang, and J. Wang, "A survey on U-shaped networks in medical image segmentations," *Neurocomputing*, vol. 409, pp. 244–258, 2020.

BMC Medical Imaging

RESEARCH

Open Access

Brain tumour segmentation based on an improved U-Net

Ping Zheng[1*], Xunfei Zhu[2] and Wenbo Guo[1]

Abstract

Background: Automatic segmentation of brain tumours using deep learning algorithms is currently one of the research hotspots in the medical image segmentation field. An improved U-Net network is proposed to segment brain tumours to improve the segmentation effect of brain tumours.

Methods: To solve the problems of other brain tumour segmentation models such as U-Net, including insufficient ability to segment edge details and reuse feature information, poor extraction of location information and the commonly used binary cross-entropy and Dice loss are often ineffective when used as loss functions for brain tumour segmentation models, we propose a serial encoding–decoding structure, which achieves improved segmentation performance by adding hybrid dilated convolution (HDC) modules and concatenation between each module of two serial networks. In addition, we propose a new loss function to focus the model more on samples that are difficult to segment and classify. We compared the results of our proposed model and the commonly used segmentation models under the IOU, PA, Dice, precision, Hausdorf95, and ASD metrics.

Results: The performance of the proposed method outperforms other segmentation models in each metric. In addition, the schematic diagram of the segmentation results shows that the segmentation results of our algorithm are closer to the ground truth, showing more brain tumour details, while the segmentation results of other algorithms are smoother.

Conclusions: Our algorithm has better semantic segmentation performance than other commonly used segmentation algorithms. The technology we propose can be used in the brain tumour diagnosis to provide better protection for patients' later treatments.

Keywords: Brain tumour, Dice loss, Encoding–decoding, HDC, Segmentation, U-Net

Background

Tumours have always been a feared disease; brain tumours have an incidence rate of 1.5% and an alarming 3% mortality rate in the population and are feared because of their extremely high incidence and mortality rate [1]. A brain tumour is a cancer of the brain tissue that is formed when the brain tissues become cancerous or metastasize to other tissues in the skull. With medical imaging technology development, imaging technology has gradually been applied to tumour detection. Initially, computed tomography (CT) technology was used for detection, but with the development of magnetic resonance physics and then the combination with the theory and technology of digital image reconstruction, magnetic resonance imaging (MRI) is slowly taking shape because it does not cause ionizing radiation damage to the body, and many imaging parameters have gradually become the mainstream of medical brain tumour detection [2]. However, most of the current clinical brain tumour diagnoses are based on the clinician's experience. The method of manually segmenting, diagnosing and annotating

*Correspondence: zp.clouds@gmail.com

[1] Anhui University of Science and Technology, Anhui 232001, China
Full list of author information is available at the end of the article

tumour images is inefficient and demanding for image analysts, and it is easy to miss the best treatment window for patients [3]. Therefore, how to efficiently diagnose brain tumour images and reduce image diagnostic error has become a research direction for many researchers. Currently, deep learning-based intelligent algorithms are widely used in brain tumour analysis tasks, and CNNs are adopted by researchers for their good segmentation performance and the convenience of feature extraction [4]. However, CNNs are prone to computational redundancy when processing a large number of dense images [5]. Therefore, FCN [6], U-Net [7] and other derived algorithms based on CNNs have been proposed. However, many brain tumour segmentation algorithms still have many problems, such as the segmentation accuracy and recognition accuracy of the algorithms are not high enough, and the attention to detail is not sufficient. In this paper, we propose an improved segmentation network based on U-Net to solve these problems using a tandem encoding–decoding model and proposing a new loss function to increase the weight of samples that are difficult to classify. The experimental results show that our proposed method outperforms several other commonly used derived models based on CNNs in terms of segmentation performance and tumour recognition accuracy.

Performing image segmentation is a key problem in the computer vision (CV) field, and image segmentation generally includes semantic segmentation and instance segmentation [8]. Brain tumour segmentation in this paper uses semantic segmentation. Evaluating the segmentation ability of the semantic segmentation model needs to focus not only on the overall image segmentation but also on edge segmentation. Therefore, how to design the segmentation algorithm becomes important, and different researchers have proposed different methods for segmentation algorithm research. With the rise of neural network models and the development of deep learning, segmentation networks based on deep learning have been rapidly developed and applied. Starting from the concept of neural networks proposed by Le Cun, neural networks have been developed rapidly, and various neural network structures have begun to emerge slowly, such as AlexNet [9], VGG [10], and ResNet [11]. Although these networks have advantages in the image recognition and prediction field, the advantages in accurate semantic image segmentation are not as obvious. To change this situation, Shelhamer et al. proposed FCN, applying FCN to semantic image segmentation [6]. They achieved segmentation by replacing the fully connected layers of the network with convolutional layers, and the results showed that the semantic image segmentation outperformed the other convolutional neural networks (CNNs). The reason is that full convolutional networks (FCNs) require a high data volume, and such brain tumour images are relatively few and precious in medicine. To solve this problem, Ronnerberger et al. modified the fully convolutional network by adopting transposed convolution, upsampling, and fusing context features and detail features to form U-Net, which can obtain enough data features with few brain images, and the segmentation effect is significantly better than that of a fully convolutional network (FCN). However, there are still problems of incomplete information and low segmentation accuracy when performing brain tumour segmentation. To solve the remaining problems of the U-Net network, Alom proposed a recursive neural network based on U-Net and a recursive residual convolution neural network based on the U-Net model [12]. Zhang et al. used the U-Net extended path to design residual connections and proposed a depth residual U-Net for image segmentation [13]. Milletari proposed a 3D U-Net model, which uses a 3D convolution kernel to expand the original U-Net structure and then adds residual units to further modify the original U-Net structure [14]. Salehi used an auto context algorithm to enhance U-Net to improve the segmentation effect [15]. Zhou et al. used the nesting method to replace the original connection method [16]. Wanli Chen, Yue Zhang et al. proposed a stacked U-Net with a bridging structure to address the problem of increasing training difficulty as the number of layers of the network increases [17]. The above segmentation model can only segment images but cannot grade segmented tumours. To achieve this clinical need, Mohamed A. Naser and M. Jamal Deen first used the trained segmentation model and MRI images for mask generation and then used a densely connected neural network classifier to classify the tumour [18].

Materials and methods

Brain tumour MRI images

The dataset was obtained from the Kaggle open source database "Brain Tumour MRI Image Classification", which contains three main types of brain tumours: glioma tumours, meningioma tumours, and pituitary tumours. The sample size of the training set containing brain tumours was 2,475, and the sample size of the test set was 289. First, we screened the dataset and selected sections with brain tumours in the sample as our experimental dataset. Then, image enhancement was carried out, and the sample size of the enhanced dataset was 2,624. Finally, manual labelling of the enhanced sections was completed with the help of graduate students from the medical college. The labelled images were reassigned according to the 10:1 ratio of the training set and test set.

Encoding module and decoding module

The SCU-Net proposed in this paper consists of two encoding modules and two decoding modules. The VGG16 net and HDC model chosen in this paper are used as the basic framework of the encoding module. Most neural network models use maximum pooling to reduce the network volume and highlight the main features when performing feature extraction. This method may ignore the segmentation details and lose the spatial location information of the main features. However, brain tumour cutting requires accuracy to the millimetre or micron level, which requires us to obtain more detailed features and minimize feature loss in training. We know that larger convolutional kernels may capture more positional information than smaller convolutional kernels because a larger receptive field can better resolve positional information [19]. Therefore, we chose hybrid dilated convolution (HDC) [20]. The HDC module can increase the receptive field, improve the ability to obtain global information, and alleviate the grid problem of dilated convolution. The hybrid dilated convolution operator is as follows:

$$\begin{aligned}
\left(F *_{l_1} k\right)(\mathbf{p}) &= \sum_{\mathbf{s}+l_1\mathbf{t}=\mathbf{p}} F(\mathbf{s})k(\mathbf{t}) \\
\left(F *_{l_2} k\right)(\mathbf{p}) &= \sum_{\mathbf{s}+l_2\mathbf{t}=\mathbf{p}} F(\mathbf{s})k(\mathbf{t}) \\
&\vdots \\
\left(F *_{l_n} k\right)(\mathbf{p}) &= \sum_{\mathbf{s}+l_n\mathbf{t}=\mathbf{p}} F(\mathbf{s})k(\mathbf{t})
\end{aligned} \tag{1}$$

In the formula, l_1 to l_n are hybrid dilated convolution operators with different dilation factors, and the maximum common divisor of L1 to Ln is not greater than 1. $F : \mathbb{Z}^2 \to \mathbb{R}$ is a discrete function. $\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$ and $k : \Omega_r \to \mathbb{R}$ are discrete filters.

We use multiple convolution blocks with different expansion rates and connect them in the same way. Each hybrid dilated convolution and a ReLU function form an HDC module, and three HDC modules with different expansion rates form an HDC module group. We replace the ordinary convolution of each layer in the two VGG encoding modules with an HDC module group. The dilation rates are selected as 1, 2 and 3 to satisfy the formula, $M_i = \max\left[M_{i+1} - 2r_i, M_{i+1} - 2(M_{i+1} - r_i), r_i\right]$, where M represents the maximum distance between two nonzero values. The distribution of the dilation rate into the sawtooth wave pattern helps the top layer obtain more information while keeping the receiving field unchanged compared with the original configuration in which the dilation rate is the same [20]. The encoding network structure is shown in Fig. 1, which is divided into 5 layers. Each layer is composed of a group of dilated convolution blocks and maximum pooling. Each group of dilated convolution blocks is composed of three 3×3 dilated convolutions (the dilation rate of dilated convolution is 1, 2, and 3), three BN layers, and three ReLU functions. The input image is $512 \times 512 \times 1$, and the image under the RGB channel is obtained through conversion. Then, a 64-channel 512×512 feature map is obtained through a dilated convolution block with a convolution kernel size of 3×3. After the first layer HDC module group and downsampling, a feature map with 128 channels of 256×256 size is obtained. Through the second layer HDC module group and downsampling, a 128×128 feature graph of 256 channels is obtained. Through the third layer HDC module group and downsampling, the 512 channel number and 64×64 size feature graph are obtained. Through the HDC module group and downsampling in the fourth layer, the 512 channel number and 32×32 size feature map are obtained. Then, these feature graphs are introduced into the decoding network and the second encoding network. SCU-Net using the HDC module can better solve the chessboard effect of dilated convolution, increase the receptive field of the coding network, and improve the capture of the details of the input image, the edge information, and the relative position information of the tumour in the input image.

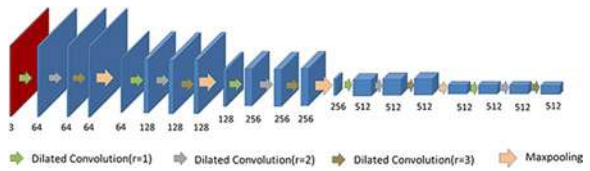


Fig. 1 SCU-Net backbone network

The decoding module of SCU-NET is composed of multiple convolutions and transposed convolutions. The information of the next layer is first expanded by a 3×3 transposed convolution of pixels and then concatenated and convolved with the feature map of the same layer. If it is in the second decoding net, it also needs to be spliced and convolved with the image of the same layer after the first decoding, then upsampled and concatenated with the upper layer, and repeatedly upsampled, concatenated and convolved, thus continuously recovering pixels until the pixels are the same as the original image, and finally convolved again, so that the number of channels is the number of the desired classification. The specific concatenation method and other details are introduced in Series Section and Concatenation.

Such a decoding method combining all the information from the previous layers and the serial network can obtain more contextual semantic information and

effectively extract the feature information of the same layers many times and then obtain more accurate prediction results and segmentation capability.

Series section and concatenation

The complete architecture of SCU-Net connects two encoding–decoding modules in series and then bridges each layer for feature sharing purposes. Several existing works show that the interactions between global features or contexts help to perform semantic segmentation, and we experimented with two bridging approaches, a pixel summing approach and a channel concatenation approach. Finally, we used the concatenation approach to connect the different modules.

Figure 2 shows the overall architecture of SCU-Net, which is composed of two encoding–decoding structures in series and has many concatenation operations between two series to add the interactions between two encoding–decoding nets. As shown in Fig. 2, we defined the feature obtained from the input image through the HDC module group with a convolution kernel size of 3×3 in the first layer of the first encoding structure as feature 1 and the feature obtained from the image downsampling in the previous layer through the HDC module group with a convolution kernel size of 3×3 in the second layer as feature 2, and the other feature maps are defined in the same order. We defined the matrix obtained by transposed convolution with a convolution kernel size of 4×4 and convolution operation with a convolution kernel size of 3×3 in the first layer of the first decoding structure as up1, and the matrix named up2 is finally obtained by concatenating with the image upsampled in the next layer and two convolutions in the second layer. The other matrices after upsampling are defined in the same order. In the first layer of the second encoding structure, up1 passes through the HDC module group with a kernel size of 3×3 to obtain feature6. In the second layer, feature7' is obtained through max-pooling with a kernel size of 2×2 and the HDC module group with a kernel size of 3×3. In the third layer, feature8' is obtained through downsampling of the upper layer feature and passing through the HDC module group with a kernel size of 3×3. The definitions of other layers are similar to those above. In the first layer of the second decoding structure, the upsampling results of the next layer are skip-connected to feature 6 and then convolved twice with a kernel size of 3×3 to obtain up6. In the second layer, the image obtained by upsampling the next layer and two convolutions with a kernel size of 3×3 is called up7'. In the third layer, the image obtained by upsampling the next layer and two convolutions with a kernel size of 3×3 is called up8'. The definitions of other layers are similar to those above.

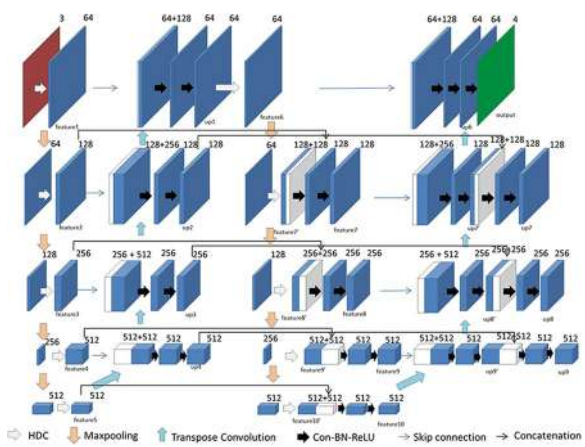


Fig. 2 Overall architecture of SCU-Net

The two serial encoding–decoding structures are connected using many concatenations. The specific connection mode is as follows: feature 5 and feature10' of layer 5 perform a concatenation operation, and then the number of channels of the image is recovered by two 3×3 convolutions to obtain feature10 with 512 channels. Feature 4 and feature 9' of the fourth layer perform concatenation and then recover the number of channels by two 3×3 convolutions to obtain feature 9 with 512 channels. Up4 and up9' perform concatenation and then recover the number of channels by two 3×3 convolutions to obtain up9 with the channel of 512. Feature3 of the third layer and feature8' perform concatenation and then recover the number of channels by two 3×3 convolutions to obtain feature8 with the channel of 256. Up3 and up8' perform concatenation and then recover the number of channels by two 3×3 convolutions to obtain up8 with the channel of 256. Feature 2 of the fourth layer and feature 7' perform concatenation and then recover the number of channels by two 3×3 convolutions to obtain feature 7 with 128 channels. Up2 and up7' perform concatenation and then recover the number of channels by two 3×3 convolutions to obtain up7 with 128 channels.

Unlike stacked U-Net, SCU-Net directly uses the output of the previous encoding–decoding structure as the input of the next encoding–decoding structure, maximizing the use of the image information after the previous decoding, avoiding structural redundancy, and being more conducive to the extraction of key information and mining. Linking the same layers of two encoding–decoding structures through concatenation can deepen the interconnection between the two encoding–decoding structures, strengthen semantic information sharing between two encoding–decoding networks, reduce feature redundancy, and accelerate learning speed. In addition, performing

secondary feature extraction of the previous information can obtain richer semantic information and thus can obtain a better segmentation effect.

Focal dice loss function

This paper proposes a new loss function called focal Dice loss. It combines the advantages of focal loss and log-cosh Dice loss and improves the disadvantage that log-cosh Dice loss can only assign the same weight to all samples so that the loss function can assign more weight to the samples that are difficult to segment to enhance the model's learning ability for the sample.

The Dice coefficient is a commonly used performance metric for segmentation tasks and thus has also been modified as a loss function to obtain higher segmentation performance, but since the Dice coefficient loss is a nonconvex function, the training process may not achieve the desired results; therefore, Shruti Jadon et al. proposed the log-cosh Dice loss to solve the problem of the nonconvex loss function by adding smoothing through Lovsz expansion [21]. The formula is as follows:

$$\cosh x = \frac{e^{x} + e^{-x}}{2} \tag{2}$$

$$L_{lc-dce} = log(cosh(\,DiceLoss\,)) \tag{3}$$

Although log-cosh Dice loss accounts for the nonconvexity problem, the same loss coefficient is used for different segmentation samples. It is not conducive to the network's learning of segmentation samples with different training difficulties. We improve the log-cosh Dice loss so that it can achieve adaptive changes for different samples by reducing the weights of easily segmented samples and adding the weights of difficult segmented samples, making the model focus more on the hard-to-segment samples during training. In addition, the improved log-cosh Dice loss and focal loss are combined so that the loss function can focus on both model classification ability and model segmentation ability. The loss function we propose is called focal Dice loss, as shown in formula (4). ω_1 and ω_2 are used to adjust the weights between the focal loss and the improved log-cosh Dice loss.$(DiceLoss)^{\gamma}$ adaptively adjusts the weight of log-cosh Dice loss in the focal Dice loss. If the log-cosh Dice loss of the current sample is large, which means the segmentation effect is poor, then $(DiceLoss)^{\gamma}$ increases. If the log-cosh Dice loss of the current sample is small, which means that the segmentation effect is good, excessive weight is not needed, and the formula automatically decreases $(DiceLoss)^{\gamma}$ to weaken the dependence on the log-cosh Dice loss.

$$\begin{aligned} Focal - DiceLoss &= \omega_1(FocalLoss) \\ &+ \omega_2(DiceLoss)^{\gamma} log(cosh(\,DiceLoss\,)) \end{aligned} \tag{4}$$

Evaluation metrics

TO verify the scientific nature of the research in this paper, the evaluation index of image semantic segmentation is selected for evaluation. It includes MIoU, MPA, precision, Dice, accuracy, Hausdorff, and ASD.

Mean-intersection-over-union (MIoU), which is the average of the ratio of intersection and merge of all categories calculated and is a common standard metric function for semantic segmentation, is as follows:

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{5}$$

where k denotes the category, i denotes the true value, j denotes the predicted value, p_{ij} denotes the number of pixels that predict i to j, p_{ji} denotes the number of pixels that predict j to i, and p_{ii} denotes the number of correctly predicted pixels.

MPA is the average of PA. PA denotes the ratio of the number of pixels correctly predicted to the total number of pixels. MPA denotes the cumulative averaging of each category, as follows:

$$MPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \tag{6}$$

In the formula, TP indicates that both the predicted and real values of the sample are positive, FP indicates that the predicted value of the sample is positive and the real value of the sample is negative.

The Dice coefficient is a similarity measurement function that judges the similarity of two samples. The greater the similarity between the two input sets, the greater the Dice coefficient, and the more accurate the segmentation model. It is one of the important indicators for image segmentation, and the specific formula is as follows:

$$Dice = \frac{2TP}{2TP + FP + FN} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{7}$$

Y represents the predicted target, and X represents the ground truth.

Accuracy is our most common evaluation metric, which refers to the ratio of the number of samples that are scored correctly to the number of all samples, and the higher the accuracy is, the better the model effect.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

TN indicates that both the predicted and real values of the sample are false. FN denotes that the predicted value of the sample is false and the real value of the sample is true.

The Hausdorff distance is used to calculate the distance between the real value boundary and the predicted value boundary, and a smaller distance between the two indicates a higher segmentation accuracy. The specific formula is as follows:

$$HD = \max\left\{\vec{d}_H(A,B), \vec{d}_H(B,A)\right\} \tag{9}$$

The average surface distance (ASD) is the average of all distances from a point on the object boundary to the GT boundary, and the specific formula is as follows:

$$ASD = \frac{1}{|B_{AS}|}\sum_{x \in B_{AS}} d(x, B_{GT}) \tag{10}$$

Result

The experiment is divided into three parts. First, we modify the convolution part of the UNetVGG16 backbone network by replacing the original convolution block with an HDC block and compare it with the original UNetVGG16 model. Then, we modify the loss function of UNetVGG16, replace the original cross-entropy loss function with our proposed focal Dice loss, and compare it with the original UNetVGG16 model. Finally, we compare the performance capability of our proposed SCU-Net with several commonly used semantic segmentation models on our brain tumour dataset.

Experimental environment

The framework used in the experiments is PyTorch, and the specifications of the machine are as follows: graphics card: Tesla P40; video memory: 22 G; CPU: Intel(R) Xeon(R) CPU E5-2680 v4; memory: 440 G, cores: 56. We optimize the model using the Adam optimizer and adjust the learning rate using the cosine annealing function by setting the cosine annealing function.

UNetVGG16 + HDC

To verify the effectiveness of the UNetVGG16 model using the HDC module, we compare UNetVGG16 with the HDC added with UNetVGG16. We replace the convolution in the original UNetVGG16 with the HDC module, train 25 epochs, and set the dilation rates of the three dilated convolutions of the HDC module to $r1 = 2$, $r2 = 3$, $r3 = 4$; the *padding* to $padding1 = 2$, $padding2 = 3$, $padding3 = 4$. The experimental results are shown in Table 1. As shown in Table 1, the results of HDC under each index are better than those of UNetVGG16. The index with the most improvement is Hausdorf95 with an increase of 27.08 percentage points, followed by MIoU with an increase of 9.51 percentage points.

UNetVGG16 + focal dice loss

After the validity of the HDC module was proven, we removed the HDC module and replaced the cross-entropy loss function in UNetVGG16 with the proposed focal Dice loss. The focal Dice loss weight was set to 4:1. As seen in Table 1, the use of the proposed focal Dice loss greatly improved, or reduced, the values of most performance indicators. MIoU and Hausdorf95 changed the most, with MIoU increasing by 9 percentage points and Hausdorf95 decreasing by 14.5 percentage points.

SCU-Net

Finally, we use SCU-Net to carry out 50 iterations, and the initial learning rate is set to 10^{-4}. The weight of the focal Dice loss function was set to 4:1. For UNetVGG16 and UNetResNet50, we use the weights pretrained by VGG16 and ResNet50 on the VOC dataset for migration learning. The initial learning rate is set to 10^{-4} in the freezing phase and 10^{-5} in the unfreezing phase. For U-Net, DeepLabv3ResNet50, and FCN8s, we directly conduct 50 iterations for training, and the initial learning rate is set as 10^{-4}. Table 2 shows the performance of SCU-Net and several commonly used segmentation algorithms under different indices in the same tumour dataset. It can be seen in the figure that all indices of

Table 1 Performance of UNetVGG16 after adding different modules

Networks \ Evaluations	MIoU (%)	MPA (%)	MPrecision (%)	MDice (%)	Hausdorf95 (mm)	ASD (mm)
UNetVggl6	68.3	78.96	81.04	79.87	52.87	1.99
HDC + UNetVggl6	77.81	84.57	89.75	86.9	25.79	0.49
Focal-Dice Loss + UNet-Vggl6	77.01	83.77	89.27	86.39	38.29	1.29

Table 2 Performance comparison between Scu-Net and several commonly used partition networks

Networks \ Evalustions	MIoU (%)	MPA (%)	MPrecision (%)	MDice (%)	Hausdorf95 (mm)	ASD (mm)
UNetVgg16	84.22	90.31	92.06	91.17	27.35	0.85
UNetResNet50	62.08	67.67	86.31	74.6	59.79	0.95
UNet	76.26	79.77	93.76	85.9	37.37	0.42
Deep1abv3ResNet50	69.48	79.5	82.58	80.82	44.01	1.07
FCN8s	81.16	84.72	94.52	89.23	30.09	0.74
SCU-Net (Ours)	86.8	90.74	94.63	92.62	17.71	0.37

SCU-Net are ahead of other networks and have the best performance, followed by UNetVGG16 and the balanced performance of all indices, followed by FCN8s. U-Net and DeepLabv3ResNet50 perform poorly. They do not achieve good segmentation accuracy and classification accuracy.

Figure 3 shows the comparison of the segmentation results and the ground truth of 6 different types of tumours by different models, in which the red region represents glioma tumours, the green region represents meningioma tumours, and the yellow region represents pituitary tumours. It can be seen in the figure that the first segmentation result of UNetVGG16 is quite different from the ground truth. The segmentation results of other classes by UNetVGG16 are basically consistent with the ground truth, but the edges are still too smooth, and the segmentation details of the edges are not ideal. Some glioma tumours are incorrectly predicted as meningioma tumours in the second segmentation figure of UNetResNet50, and the segmentation result is a triangle, which is quite different from the ground truth. The second segmentation result graph of U-Net almost does not segment the tumour, so the effect was poor. The last segmentation map of DeepLabv3ResNet50 has nearly 50% of the regions incorrectly classified, while the segmentation results of other segmentation maps are obviously too smooth, although the categories are correctly predicted. The effect of the second segmentation map of FCN8s is poor, which is quite different from the ground truth. The segmentation result of SCU-Net (Ours) is the best, there is almost no error prediction, and the edge segmentation effect is obviously better than other models. The experimental results show that SCU-Net has good robustness on the brain tumour dataset presented in this paper.

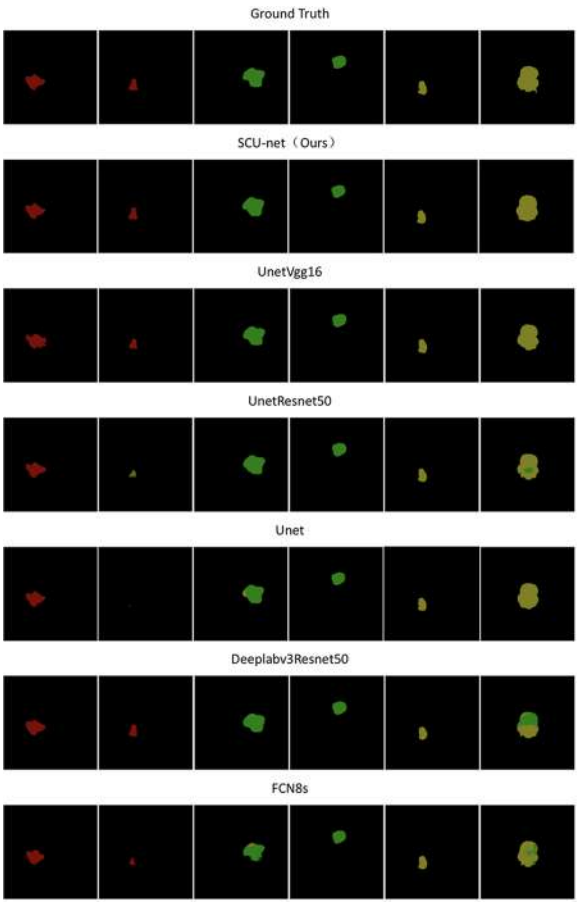


Fig. 3 Comparison of segmentation results of different types of tumours by different networks

Discussion

UNetVGG16 + HDC

To verify whether UNetVGG16 with the HDC module can capture more detailed information, we compare UNetvGG16 with HDC with the original UNetVGG16. It can be seen in Table 1 that HDC + UNetVGG16 has great improvements in both segmentation ability and classification accuracy. This is because the HDC module enlarges the receptive field of the model and solves the problem of incomplete information capture caused by the traditional expanded convolution chessboard phenomenon so that the algorithm can better extract the characteristics of the image. Therefore, it is a good choice to use HDC in the backbone network of

UNetVGG16, which can improve the model's classification accuracy and segmentation accuracy.

UNetVGG16 + focal dice loss

To verify whether our proposed focal Dice loss can specifically learn samples that are difficult to segment and classify to improve the overall performance of the algorithm, we designed a comparative experiment between UNetVGG16 + focal Dice loss and UNetVGG16, as shown in Table 1. The experimental results show that the proposed focal Dice loss contributes to the segmentation network performance and greatly improves tumour recognition accuracy and segmentation performance. We accelerate the training of difficult training samples by increasing the loss weight of difficult segmentation and classification to improve the model performance, which means that we can better deal with brain tumour samples that are more difficult to train in the training set to effectively solve the problems of unbalanced tumour category samples and different qualities of tumour sample images.

SCU-Net

Table 2 shows that our SCU-Net is different from U-Net and the other four commonly used segmentation algorithms. It can be seen in the table that SCU-Net has the best performance under most indicators, and its performance is far better than that of U-Net. Figure 3 shows the segmentation results of SCU-Net, U-Net and several other commonly used segmentation algorithms. It can be seen that the effect of SCU-Net is closest to the ground truth, and the detail processing is also better than the other algorithms. According to the experiment, it can be inferred that although FCN8s adopt 8 times upsampling, which is much better than 32 times, it still lacks details and is not sufficient to achieve a high segmentation effect. UNetResNet50 and DeepLabv3ResNet50 easily produce gradient disappearance and other problems because the backbone network model is too deep. The tumour images are composed of simple textures and shapes and the most typical features. If ResNet50 is used, it may cause feature redundancy, which affects the final result, so VGG16 as the SCU-Net backbone network is the best choice. Through serial operation, SCU-Net not only does not result in feature redundancy but also uses two decoding networks for feature extraction twice, as well as the fusion of features and pixels, which improves the semantic segmentation performance of the network.

Conclusion

In this paper, we proposed an improved U-Net algorithm, called SCU-Net, for segmenting brain tumours. We operate two U-Net models with VGG16 as the backbone in tandem and perform feature splicing and decoding module splicing at each layer so that the two encoding–decoding blocks before and after can form a tighter connection to obtain more semantic information, reduce feature redundancy, and further improve the generalization ability of the model. Since location information is extremely important for brain tumour category classification, we add another HDC module to the U-Net encoding network to obtain a larger receptive field to capture more location information. In addition, the proposed focal Dice loss enables the model to consistently focus not only on pixel classification accuracy but also on segmentation performance during training and focus more on the samples that are difficult to classify and divide. We compared SCU-NET with commonly used brain tumour segmentation models under 6 indicators. Experimental results show that the proposed method can significantly improve target segmentation and tumour prediction performance.

Acknowledgements

Not applicable.

Author contributions

PZ designed the algorithm and is the main contributor to the manuscript. XZ participated in the analysis and discussion of the experiment and the preparation of the manuscript. WG participated in the preparation of the manuscript and the preliminary material preparation. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The data in this article adopt Kaggle's official open source dataset. Download address: https://www.kaggle.com/datasets/iashiqul/brain-tumor-mri-image-classification.

Declarations

Ethics approval and consent to participate

In this paper, all methods and data were analysed in accordance with relevant guidelines and regulations.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Anhui University of Science and Technology, Anhui 232001, China. [2]Yunnan University, Yunnan 650500, China.

Received: 28 July 2022 Accepted: 8 November 2022
Published online: 18 November 2022

References

1. Mayer G, Vrscay E. Self-similarity of Fourier domain MRI data. Nonlinear Anal Theory Methods Appl. 2009;71(12):e855–64.
2. Mohan G, Subashini MM. MRI based medical image analysis: survey on brain tumor grade classification. Biomed Signal Process Control. 2018;39:139–61.
3. Hu HX, Mao WJ, Lin ZZ, Hu Q, Zhang Y. Multimodal brain tumor segmentation based on an intelligent UNET-LSTM algorithm in smart hospitals. ACM Trans Internet Technol. 2021;21(3):14.
4. Hao K, Lin S, Qiao J, Tu Y. A generalized pooling for brain tumor segmentation. IEEE Access. 2021;9:159283–90.
5. Yang T, Song J, Li L, Tang Q. Improving brain tumor segmentation on MRI based on the deep U-net and residual units. J X-Ray Sci Technol. 2020;28(1):95–110.
6. Long J, Shelhamer E, Darrell T. IEEE: fully convolutional networks for semantic segmentation. In: Proceedings of IEEE conference on computer vision and pattern recognition. Boston, MA: IEEE; 2015: 3431–3440.
7. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: 18th Proceedings of international conference on medical image computing and computer-assisted intervention, vol. 9351. Munich, Germany. Springer International Publishing Ag; 2015: 234–241.
8. Yu H, Yang Z, Tan L, Wang Y, Sun W, Sun M, Tang Y. Methods and datasets on semantic segmentation: a review. Neurocomputing. 2018;304:82–103.
9. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Proceedings of advances in neural information processing systems. 2012;25.
10. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014.
11. He KM, Zhang XY, Ren SQ, Sun J. IEEE: Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition. Seattle, WA; 2016:770–778.
12. Alom MZ, Hasan M, Yakopcic C, Taha TM, Asari VK. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. arXiv preprint arXiv:1802.06955. 2018.
13. Zhang Z, Liu Q, Wang Y. Road extraction by deep residual u-net. IEEE Geosci Remote Sens Lett. 2018;15(5):749–53.
14. Milletari F, Navab N, Ahmadi SA. IEEE: V-Net: fully convolutional neural networks for volumetric medical image segmentation. In: 4th IEEE international conference on 3D vision (3DV). Stanford University, Stanford, CA: IEEE; 2016: 565–571.
15. Salehi SSM, Erdogmus D, Gholipour A. Auto-context convolutional neural network (auto-net) for brain extraction in magnetic resonance imaging. IEEE Trans Med Imaging. 2017;36(11):2319–30.
16. Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J. Unet++: redesigning skip connections to exploit multiscale features in image segmentation. IEEE Trans Med Imaging. 2019;39(6):1856–67.
17. Chen W, Zhang Y, He J, Qiao Y, Chen Y, Shi H, Wu EX, Tang X. Prostate segmentation using 2D bridged U-net. In: 2019 International joint conference on neural networks. IEEE; 2019:1–7.
18. Naser MA, Deen MJ. Brain tumor segmentation and grading of lower-grade glioma using deep learning in MRI images. Comput Biol Med. 2020;121: 103758.
19. Islam MA, Jia S, Bruce ND. How much position information do convolutional neural networks encode? arXiv preprint arXiv:2001.08248;2020.
20. Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, Cottrell G. IEEE: understanding convolution for semantic segmentation. In: 18th IEEE winter conference on application of computer vision. Nv; 2018:1451–1460.
21. Jadon S. A survey of loss functions for semantic segmentation. In: IEEE conference on computational intelligence in bioinformatics and computational biology. IEEE;2020:1–7.

Publisher's Note

frontiers | Frontiers in Big Data

TYPE Methods
PUBLISHED 06 January 2023
DOI 10.3389/fdata.2022.1080715

OPEN ACCESS

EDITED BY
Xiaoqian Wang,
Purdue University, United States

REVIEWED BY
Qi Huang,
The University of Utah, United States
Mengting Liu,
Sun Yat-sen University, China

*CORRESPONDENCE
Haoteng Tang
✉ haoteng.tang@pitt.edu
Liang Zhan
✉ liang.zhan@pitt.edu

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Big Data

RECEIVED 26 October 2022
ACCEPTED 06 December 2022
PUBLISHED 06 January 2023

CITATION
Fu X, Sun Z, Tang H, Zou EM, Huang H, Wang Y and Zhan L (2023) 3D bi-directional transformer U-Net for medical image segmentation. *Front. Big Data* 5:1080715. doi: 10.3389/fdata.2022.1080715

3D bi-directional transformer U-Net for medical image segmentation

Xiyao Fu[1], Zhexian Sun[2], Haoteng Tang[1]*, Eric M. Zou[3], Heng Huang[1], Yong Wang[2,4,5,6] and Liang Zhan[1]*

[1]Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, United States, [2]Department of Biomedical Engineering, Washington University in St. Louis, St. Louis, MO, United States, [3]Montgomery Blair High School Maryland, 51 University Blvd E, Silver Spring, MD, United States, [4]Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, United States, [5]Department of Obstetrics and Gynecology, Washington University in St. Louis, St. Louis, MO, United States, [6]Department of Radiology, Washington University in St. Louis, St. Louis, MO, United States

As one of the popular deep learning methods, deep convolutional neural networks (DCNNs) have been widely adopted in segmentation tasks and have received positive feedback. However, in segmentation tasks, DCNN-based frameworks are known for their incompetence in dealing with global relations within imaging features. Although several techniques have been proposed to enhance the global reasoning of DCNN, these models are either not able to gain satisfying performances compared with traditional fully-convolutional structures or not capable of utilizing the basic advantages of CNN-based networks (namely the ability of local reasoning). In this study, compared with current attempts to combine FCNs and global reasoning methods, we fully extracted the ability of self-attention by designing a novel attention mechanism for 3D computation and proposed a new segmentation framework (named 3DTU) for three-dimensional medical image segmentation tasks. This new framework processes images in an end-to-end manner and executes 3D computation on both the encoder side (which contains a 3D transformer) and the decoder side (which is based on a 3D DCNN). We tested our framework on two independent datasets that consist of 3D MRI and CT images. Experimental results clearly demonstrate that our method outperforms several state-of-the-art segmentation methods in various metrics.

KEYWORDS

semantic segmentation, COVID, lung, placenta, transformer, 3D UNet, CT, MRI

1. Introduction

In the recent few years, deep convolutional neural networks (DCNNs) (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2016; Badrinarayanan et al., 2017; Huang et al., 2020; Pan et al., 2020) have achieved considerable progress in medical image segmentation (Long et al., 2015; Noh et al., 2015; Chen L.-C. et al., 2018; Tokunaga et al., 2019; Liu et al., 2022; Zhang et al., 2022). However, limited to the local receptive field of the convolutional filter, DCNN-based frameworks are incapable of capturing long-range dependencies from global features for semantic segmentation. To tackle this, several

strategies can be considered. The first is to use the dilated convolution operation to enlarge the size of the receptive field of the convolutional filter (Yu and Koltun, 2015; Yang et al., 2017; Zhang et al., 2017; Liu et al., 2021). However, this enlarged local receptive field is still limited by the size of dilation. Another solution is to model the feature map as graph structures and investigate the long-range dependencies through the message passing mechanism of different graph learning models (e.g., graph convolution networks) (Li and Gupta, 2018; Chen et al., 2019; Li et al., 2020; Jia et al., 2021). Although these graph learning models have shown great potential in enhancing the global reasoning ability of DCNNs, they have very high requirements for computation and memory due to the constructed large-size graphs.

The attention mechanism (Hochreiter and Schmidhuber, 1997; Vaswani et al., 2017) is a computation scheme that tries to generate representations *via* different types of global features at each step. Since attention can be regarded as the conversion and transformation among the query (q), key (k), and value (v) triplet, attention computation is to generate the q based on the combination of the k–v pair. As it is natural to integrate a cycling computation in recurrent cells, traditional attention mechanisms are integrated within recurrent neural networks (e.g., Hochreiter and Schmidhuber, 1997; Cho et al., 2014), which inevitably impairs the efficiency of recurrent networks compared with linear/residual networks (Vaswani et al., 2017). To cope with this, Vaswani et al. (2017) proposed a transformer, a structure consisting of a series of identical encoder blocks connected with a series of identical decoder blocks, which all have no convolutional layers and are connected in a residual way. The original transformer supported by self-attention works exceptionally well in some tasks like machine translation but not in visual tasks (Chen et al., 2021). This is mainly due to the lack of convolution layers that makes the model struggle to detect local features.

For the aforementioned reasons, convolutional-based frameworks are still preferred for segmentation tasks. Although several other models (Goodfellow et al., 2014; Chen Y. et al., 2018) have been proven feasible, DCNNs remain to be one of the most effective methods. Multiple variants of DCNNs have been proposed to make the segmentation process more effective, one of the most crucial ones is the UNet (Ronneberger et al., 2015), which is a symmetric structure consisting of convolutional blocks with skip connections. These convolutional blocks have descending dimensions on the encoder side and ascending dimensions on the decoder side. However, due to the intrinsic fully convolution structure, UNet is suboptimal to relate local features to global representations with more variant distribution (Chen et al., 2021). To cope with the drawbacks of UNet, many methods have been proposed (Liu et al., 2018; Zhou et al., 2019; Diakogiannis et al., 2020; Huang et al., 2020). However, these methods are either very time-consuming or require heavy computations, which make it impossible to be applied to 3D objects.

Under such circumstances, the self-attention mechanism seems to be a nearly optimal solution. It is highly modulized and can stretch the number of self-attention cells according to the training environment. It can also train on vast datasets due to the training nature of attention. Therefore, researchers combined the transformer with convolutional layers for medical image segmentation (Li et al., 2022). On the one hand, the transformer encodes tokenized image patches from a CNN feature map as the input sequence for extracting global contexts. On the other hand, the decoder upsamples the encoded features, which are then combined with high-resolution CNN feature maps to enable precise localization.

However, this approach still has some obstacles, especially in the segmentation of 3D objects. This is partially due to transformers (Vaswani et al., 2017) requiring the input features to have temporal information. Since the self-attention does not compute with a clear direction, features have to be preprocessed with temporal info (e.g., cosine function) as input embeddings before training. Although this learning process can be seen as natural (scanning the features linearly and with order), it will restrict the performance of high-dimensional data. For example, many existing transformer approaches (Parmar et al., 2018; Huang et al., 2020; Chen et al., 2021) will cut the 3D object into 2D slice sequences to meet the temporal encoding requirement; however, the segmentation performance is actually worse because the 2D slice cutting will destroy the smoothness of the object in 3D space. Bi-directional transformer (Devlin et al., 2018) is a powerful upgrade version of transformer. It is a structure with no decoder and processes the inputs all at once with masks to create temporal/spatial continuity. However, we will show in the experiment section that bi-directional transformers can serve as a strong encoder but still struggles to get better results on 3D segmentation. To compensate for the loss of feature resolution brought by transformers, we propose 3D transformer UNet (3DTU), which employs a hybrid CNN–transformer architecture to leverage both detailed high-resolution spatial information from CNN features and the global context encoded by our new 3D bi-directional transformer module. We show that such a design allows our framework to preserve the advantages of self-attention mechanisms and also get considerably improved results on 3D image segmentation compared with previous U-Net-based or transformer-based methods. To sum up, our contributions to this article can be summarized as follows:

- We proposed a new 3D bi-directional framework to learn deep 3D features for medical image semantic segmentation.
- We designed a novel attention mechanism specifically suitable for network training and self-attention computation for 3D objects.

- We verified our new framework on multiple datasets, consisting of different imaging modalities (MRI and CT images) and different organs (placenta and lungs infected with COVID) and obtained state-of-the-art (SOTA) results. Our method beat baselines in performances on multiple metrics.

2. Related work

2.1. Fully convolutional network in medical image segmentation

Many studies have attempted to adopt convolutional networks to medical image segmentation. For example, Liu et al. (2018) presented a hybrid network consisting of both 3D CNN and 2D CNN in the brain image segmentation for Alzheimer's disease (AD) studies. Ronneberger et al. (2015) presented UNet, one of the most iconic encoder–decoder-based methods for medical image segmentation. Their method consists of convolutional blocks that have a U-shaped dimension variation. Specifically, from the input layer of the encoder to the input layer of the decoder, each block's dimension is descending. And the decoder has an ascending dimension that is matched to the encoder blocks. Such a design makes sure that the learning ability of the framework is powerful enough to find the abstract of the locality and output a global representation map. Several adjustments (e.g., Zhou et al., 2019; Huang et al., 2020) have been made to the original UNet model. For example, U-Net3+ (Huang et al., 2020) and its variations, although proved effective, still suffer from the locality-heavy learning scheme. Some researchers tried to boost the local reasoning of convolutional layers through the residual structure. For example, ResUNet (Diakogiannis et al., 2020) proposed a residual block between every two convolutional blocks on both the encoder side and decoder side as well as skip-connection between residual blocks with the same dimension between the encoder and decoder. Isensee et al. (2021) argued that the understanding of the datasets needed for training is more important than the network itself since most UNet-based moderations have achieved little progress. The authors proposed nnUNet, a robust network, that is designed based on the combination of 2D and 3D UNet. The authors also made different training configurations (normalization tricks, cropping, activation functions, etc.) based on the datasets.

2.2. Transformers

Transformers (Vaswani et al., 2017) were initially proposed for general NLP tasks and quickly gain widespread attention by beating previous most state-of-the-art results by a large margin. Devlin et al. (2018) converted the original transformer model into BERT, and introduced the called bi-directional transformers, which are proven effective again. Naturally, multiple efforts have been made to adjust the learning ability of transformers in the computer vision domain. Several variants of transformers have emerged recently. Parmar et al. (2018) presented one of the early works to adjust vanilla transformers by incorporating visual information. This model pre-processes each pixel of one image through a 1×1 convolution layer. Then, the embeddings are computed with positional embeddings before feeding into transformers for super-resolution tasks. In another attempt at visual tasks, Dosovitskiy et al. (2020) proposed Vision transformer (ViT), which presented a novel way of input embedding on visual information. It achieved state-of-the-art on ImageNet classification by directly applying transformers with global self-attention to full-sized images. Specifically, ViT flattens an image to fixed-sized pixels, which then be linearly added to positional embeddings before feeding to transformer encoders. Valanarasu et al. (2021) presented gated axial attention that creates a gated scheme to improve learning ability on the local scale.

2.3. Combination of UNet and transformer in medical image segmentation

Multiple attempts have been made to combine the UNet with transformer in both framework structure and inner encoder/decoder computation. TransUNet (Chen et al., 2021) consists of a series of transformer units as the encoder and the right half of the UNet as the decoder to generate predictions in medical image segmentation. Both the encoder and the decoder (Chen et al., 2021) are computed in a 2D scenario. Yun et al. (2021) introduced SpecTr, a framework that takes spectral normalization into the computation between convolution and attention blocks. Their methods achieved better results than the baseline when training on hyperspectral medical images. Wang et al. (2021) presented TransBTS that utilizes 3D CNN to extract input representations. UNet transformer, presented by Petit et al. (2021), replaces self-attention modules in transformer encoder/decoder cells by convolutional blocks and batch normalization computations. Another attempt is Swin-UNet (Cao et al., 2021), which instead replaces convolution blocks in the UNet-Structure network with self-attention modules. Several works follow similar methods including UNETR (Hatamizadeh et al., 2022b), SWIN UNETR (Hatamizadeh et al., 2022a), CoTr (Xie et al., 2021), nnFormer (Zhou et al., 2021), DS-TransUNet (Lin et al., 2022), UTNet (Gao et al., 2021), and PNS-Net (Ji et al., 2021). In UNETR, the authors presented a novel 3D transformer encoder and a voxel-wise loss for model training. For the positional embedding, they adopted a strategy from the Visual

transformer, which divides the 3D images into 3D patches. The decoder in their work consists of several convolutional blocks in different dimensions and skip connections to the encoder. The SWIN UNETR is proposed for 3*D* multi-modal MRI brain image studies, which is different from the SWIN UNET that is proposed for 2D images. The CoTr utilized a DeTrans-encoder with a novel attention mechanism and a CNN-based decoder. The nnFormer utilizes CNN as part of an encoder, which leverages the ability of local feature extraction of CNN structures. Moreover, it utilizes transformer structures as its decoder and the second part of its encoder. There are two differences between our 3DTU and the nnFormer. First, we utilize a CNN-based structure (i.e., the right part of 3DUNet) as our decoder. Then, we design an attention mechanism that computes the attention scores from different directions.

The aforementioned methods adjust the transformers in visual tasks by introducing their own positional embedding rules. Although these rules are to an extent useful, their performance all suffers from the slicing of 3D data to adjust the positional embeddings. In this study, positional embeddings are not needed technically, even for 3D data. We modify the multi-head attention from its original form to a refined computation scheme that fully utilizes the potentials of transformer and UNet. More importantly, our encoder is a refined bi-directional transformer, which learns the feature from three (i.e., along x, y, and z) directions simultaneously.[1]

3. Methods

We propose a 3D UNet-based framework with bi-directional transformers (named 3DTU) in this work. The self-attention mechanism in the proposed bi-directional transformers can improve the ability of generalization of the framework encoder. We will delve into the technical details in this section.

As shown in Figure 1, our proposed 3DTU is an encoder–decoder framework, where the encoder consists of two modules including a feature extraction module (see Part I in Figure 1) and a bi-directional transformer module (see Part II in Figure 1). Given a 3D image $I \in \mathcal{R}^{h\times w\times d\times c}$, where h, w, and d are the shapes of the image and c is the image channel number, the feature extraction module projects the 3D image I as a latent representation X *via* basic convolutional neural networks (CNNs). Then, the 3D bi-directional transformer cells take the latent representation X as input and yield the masked latent representation X_M by using Masked-LM (MLM) (Devlin et al., 2018) step by step. Finally, the decoder part utilizes the masked latent representations to reconstruct the segmentation predictions for loss computation.

3.1. Encoder with 3D bi-directional transformer

As aforementioned, the encoder of the 3DTU consists of two parts. The first part of the encoder is a CNN-based feature extraction module. We aim to convert the original 3D image (I) into an iso-dimensional latent cube representation ($X \in \mathcal{R}^{1\times p\times p\times p}$) *via* this module as assistance to capture the image locality for transformer modules, since the transformer module may not have enough ability to capture the image local features. We will show this point in the ablation studies. Particularly, the feature extraction module includes two convolutional layers followed by a fully-connected (FC) layer and a max-pooling layer in between the two convolutional layers. The FC layer is used to adapt the feature dimension.

The bi-directional transformer module takes the latent cube representation X as input and computes multi-head attentions with the MLM strategy (Devlin et al., 2018). Details of the bi-directional transformer module are shown in Figure 2. In general, each cell in the bi-directional transformer module generates the latent feature map X_1 by the following steps:

$$\begin{aligned} X^{'} &= Att(Norm(X)) + X, \\ X^{''} &= FF(Norm(X^{'})), \\ X_1 &= X^{'} + X^{''}, \end{aligned} \tag{1}$$

where $Att(\cdot)$ is the multi-head self-attention operation, $Norm(\cdot)$ is a 3D normalization operation, and $FF(\cdot)$ is the feed forward layer (i.e., FC layer). $+$ denotes a pixel-wise add operation. Particularly, the multi-head attention is computed as follows:

$$\begin{aligned} Att\_head_i^{x,y,z} &= SDP(Q,K,V)\times W, \\ MultiHead(Q,K,V) &= Concat(head_i^x, head_i^y, head_i^z), \end{aligned} \tag{2}$$

where $SDP(\cdot)$ is the Scaled Dot-Product Attention, W is the trainable parameters for linear projections (i.e., L_q, L_k, and L_v in Figure 2) and $Concat(\cdot)$ denotes a concatenation operation. Q, K, and V are the query-key-value triplets defined by the transformer cell. Note that our proposed attention mechanism can yield the attention score by scanning the query-key-value triplets in three different directions (i.e., along x, y, and z axes, respectively), which gain plentiful discriminative and anisotropic semantic information for the 3D image segmentation.

1 We use the term "bi-directional" by following previous studies. However, our 3DTU learns the features from three directions instead.

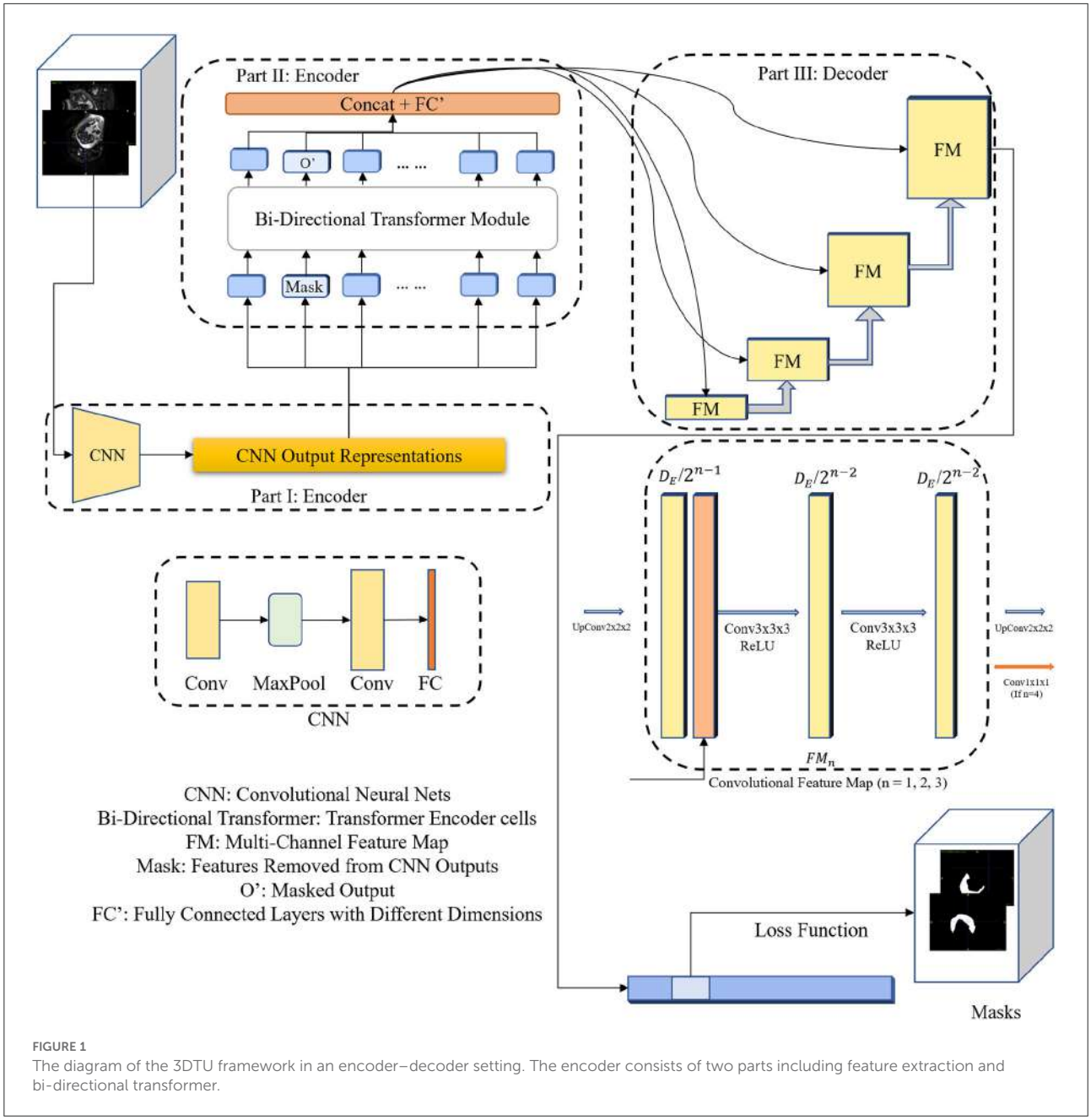


FIGURE 1
The diagram of the 3DTU framework in an encoder–decoder setting. The encoder consists of two parts including feature extraction and bi-directional transformer.

3.2. UNet-based decoder

As shown in Figure 1, we utilize convolutional blocks with ascensional dimensions in the decoder part. A residual connection is adopted between the encoder side and the decoder side. Particularly, a cascaded of multi-channel feature map (FM) blocks are integrated into the decoder part, each of which contains two $3 \times 3 \times 3$ convolutional layers and an upsampling layer. The channel number of feature maps reduces by half after each FM block. In the last FM block, instead of upsampling layer, a $1 \times 1 \times 1$ convolutional layer is used to generate final segmentation predictions.

3.3. Loss function and supervision manner

Since the MLM strategy is used in the encoder part, where a portion of image features are masked (i.e., set to 0 values)

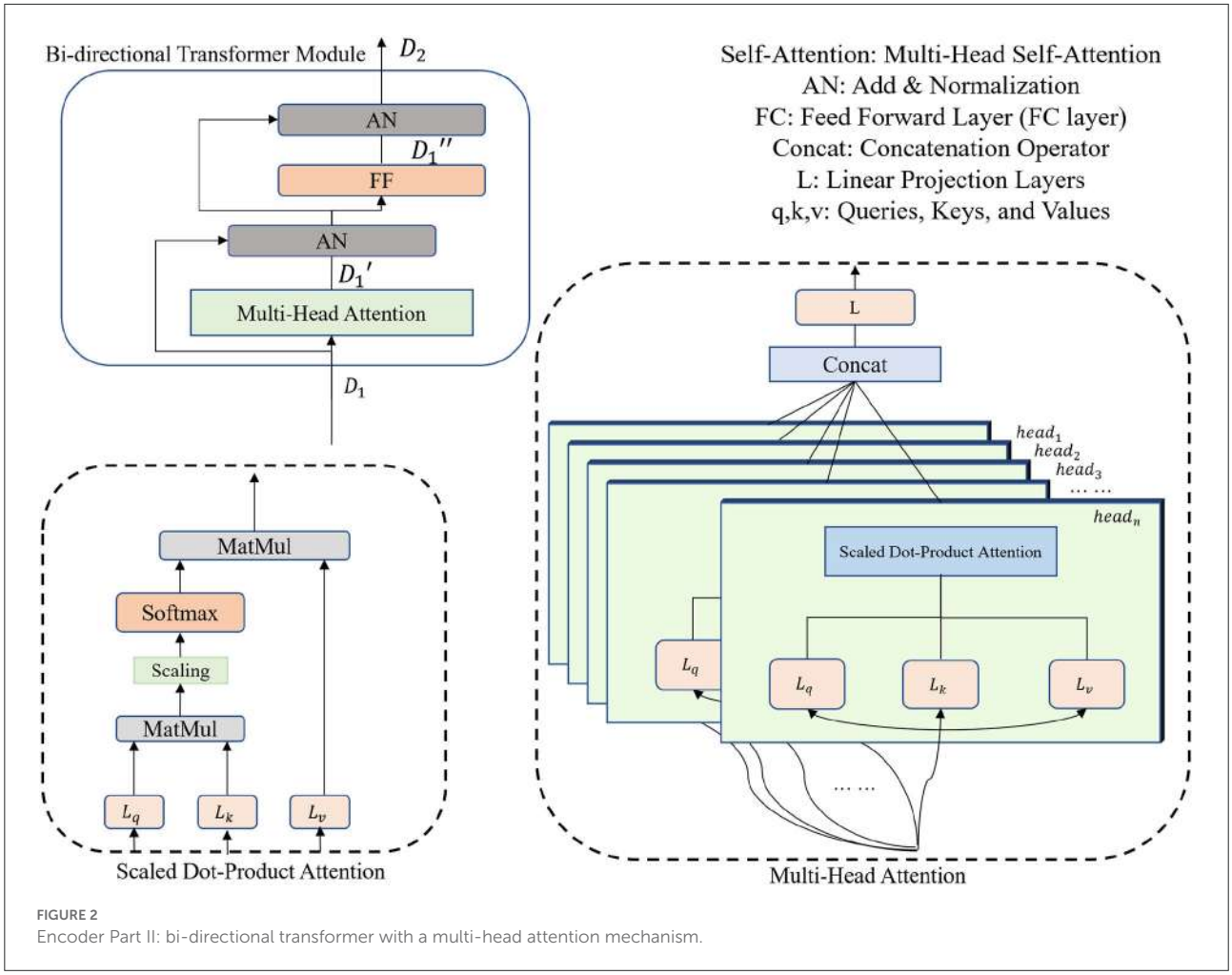


FIGURE 2
Encoder Part II: bi-directional transformer with a multi-head attention mechanism.

and the other portions remain the same. Hence, our goal is to use the uncovered portions to predict the masked portions (Devlin et al., 2018), in which the loss is only estimated based on the masked regions. Particularly, the loss function can be formulated as:

$$\mathcal{L} = \alpha \times \ell_{dice}(\hat{y}_{mask}, y_{mask}) + (1-\alpha) \times \ell_{BCE}(\hat{y}_{mask}, y_{mask}), \quad (3)$$

where $\hat{y}_{mask}$ and y_{mask} are the masked regions of segmentation prediction and ground truth, respectively. $\alpha \in [0, 1]$ is the loss weight.

4. Experiments

4.1. Datasets

We used three datasets obtained from different modalities for this study, including Placenta MRI (Placenta) dataset, COVID-19 CT lung and infection segmentation (Covid20) dataset, and Multi-Atlas Labeling Beyond the Cranial Vault (Synapse) dataset. Details of data description and preprocessing are shown below.

Placenta MRI dataset was collected from the Washington University in Saint Louis (WUSTL) (Sun et al., 2022), where all data were de-identified before processing. The data collection and related studies were approved by the Institutional Review Board at the WUSTL. A total of 81 MRI scans were collected from 46 pregnant patients (mean age = 23.91 ± 3.02 yo, mean BMI = 25 ± 3.66 at recruitment) with normal singleton pregnancy who underwent MRI during the third trimester, by a Siemens 3T VIDA scanner. Of the 46 patients, 21 patients had the single scan and 25 patients had multiple longitudinal scans. The average gestational ages (GA) during MRI scans were 34.12 ± 1.07 weeks (Min GA 28 weeks 3 days, max GA 38 weeks 6 days). T2-weighted MRI of the entire uterus was acquired with a 2D EPI sequence in the left lateral position. The MRI data has a fixed acquisition matrix of 128 × 128 × 115, and variable voxel sizes from 3 × 3 × 3 mm to 3.5 × 3.5 × 3.5 mm, up to the patient's size. Manual segmentation of the placenta

TABLE 1 Quantitative segmentation results of different methods on two datasets, where mIOU and DICE are in %.

	Placenta dataset			Covid20 dataset			Synapse dataset		
	mIOU	DICE	HD95	mIOU	DICE	HD95	mIOU	DICE	HD95
2D UNet	67.6	72.3	12.0	73.6	78.3	112.5	56.3	60.6	45.7
3D UNet	72.5	78.6	10.7	78.1	84.0	97.6	59.4	62.2	42.2
UNet++	74.5	77.1	8.2	80.3	84.6	63.0	67.1	73.7	34.0
TransUNet	73.6	80.0	7.4	83.1	89.2	45.8	70.2	77.5	31.7
ViT	72.9	79.7	8.5	84.2	89.0	70.3	65.3	67.9	36.1
nnFormer	78.3	82.1	10.2	81.0	89.9	66.2	81.8	86.6	10.6
nnUNet	78.9	83.6	8.7	90.3	91.6	59.9	84.2	89.8	16.6
3DTU (Ours)	79.8	84.0	7.2	90.5	92.0	59.4	85.0	87.3	18.4

The best results are shown in red and the second-best results are shown in blue.

regions was conducted by experienced radiologists for all MRI images.

COVID19-CT-Seg20 dataset (Covid20) contains 20 COVID-19 3D CT images, where lungs and infections were annotated by two radiologists and verified by an experienced radiologist[2] (Jun et al., 2021). We only focused on the segmentation of the COVID-19 infections in this study, since it is more challenging and important.

Multi-atlas labeling beyond the cranial vault (Synapse) dataset.[3] We use the 30 abdominal CT scans from the MICCAI 2015 Multi-Atlas Abdomen Labeling Challenge. These scans were captured during the portal venous contrast phase with variable volume sizes ($512 \times 512 \times 85$–$512 \times 512 \times 198$) and field of views (approximately $280 \times 280 \times 280\ mm^3$–$500 \times 500 \times 650\ mm^3$). The in-plane resolution varies from $0.54 \times 0.54\ mm^2$ to $0.98 \times 0.98\ mm^2$, while the slice thickness ranges from 2.5 to 5.0 mm. We report the average experimental results on eight abdominal organs (aorta, gallbladder, spleen, left kidney, right kidney, liver, pancreas, spleen, and stomach) with 5-fold validation.

4.2. Implementation details

In the pre-processing step, we simply normalized the intensities of each 3D image to zero mean and unit variance. In the training phase, we applied data augmentation techniques to reduce potential overfitting, including random rotation of the image by 90° along three dimensions and adjusting the brightness of the top 3% pixels. The training iterations were set to 10^5. We trained the model using the Adam optimizer with a batch size of 1 and synchronized batch normalization. The initial learning rate was set to $1e-2$ and was decayed by $(1 - \frac{current\_epoch}{max\_epoch})^{0.9}$. We also regularized the training with dropout in the transformer cells. All experiments are conducted using a 5-fold cross-validation, based on Pytorch 1.7.1 on a workstation with 2 NVIDIA TITAN RTX GPUs. The data division on the public Covid20 dataset is adopted by following the division strategy given by Qiu et al. (2021).

As aforementioned, our encoder consists of two parts. In the feature extraction module, we used a CNN network with two convolutional layers, one max-pooling layer, and one 1-D fully-connected layer with the direction of $x-y$ plane to z coordinate to convert the representations with the original dimension to a cube. The first convolutional layer, with a kernel size of $3 \times 3 \times 3$, embeds the input 3-D image into local representation maps, while the second convolutional layer project the local representation maps for the second part of the encoder *via* a linear transformation. The output dimension of the feature extraction module is converted (i.e., reshape) to $X \in \mathcal{R}^{1\times256\times256\times256}$. In the bi-directional transformer module, we utilize multiple transformer cells with the bi-directional self-attention mechanism. Specifically, the input embedding strategy that we adopted is Masked LM (MLM) (Devlin et al., 2018). The Masked LM has been proven to be useful within the previous BERT paper (Vaswani et al., 2017), where the image portion masked in the encoder is matched to that in the loss computation stage. Moreover, since we do not embed the data with the positional encoding in our framework, we require a way to learn the 3D representations through a certain sequence. MLM can well meet this requirement. We set the number of transformer cells as 12, 6, and 6 for Placenta, Covid20, and Synapse datasets, respectively. The number of heads within each transformer cell is 15, where each direction (i.e., $x-y$, $x-z$, and $y-z$ plane) contains five heads to compute self-attention scores. The length of each mask is set to 16, 32, and 32 for the Placenta, Covid20, and Synapse datasets, respectively. Each cube representation is divided into 16 parts in the training phase.

2 https://zenodo.org/record/3757476#.Y1NGmy1h1B1

3 https://www.synapse.org#!Synapse:syn3193805/wiki/217789

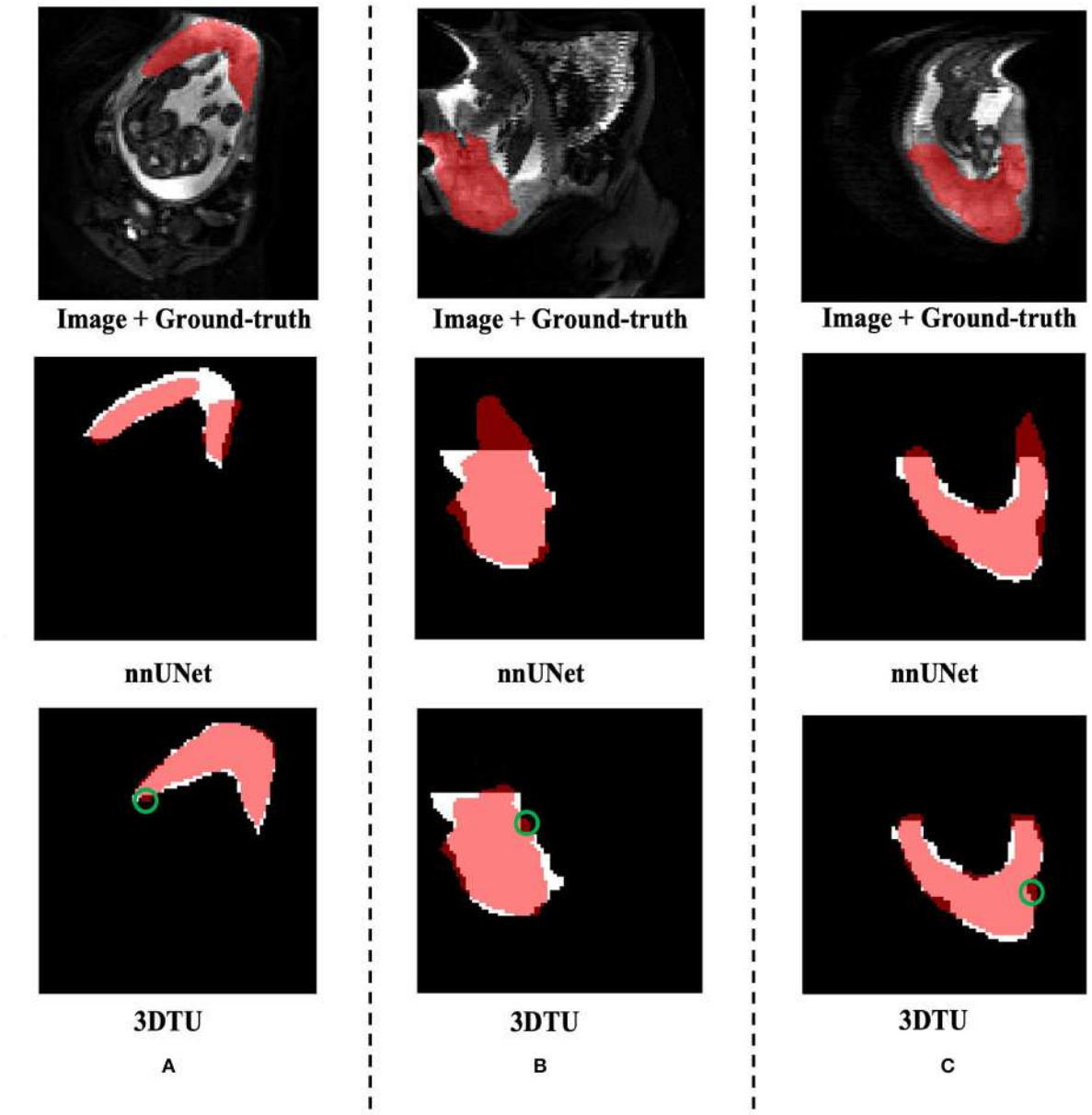


FIGURE 3
Visualization of the segmentation results on the Placenta dataset produced by our 3DTU and nnUNet. Columns **(A–C)** show the *x*–*y* plane, *y*–*z* plane, and *x*–*z* plane of 3D segmentation predictions, respectively. The true-positive regions are highlighted in pink. The false-negative regions are highlighted in red (e.g., the green circle regions in the last row). Better view with colors and zooming in.

4.3. Baseline settings and evaluation metrics

To evaluate our 3DTU's performance, we choose the following frameworks as baselines: 2DU-Net (Ronneberger et al., 2015), 3D U-Net (Çiçek et al., 2016), UNet++ (Zhou et al., 2019), TransUNet (Chen et al., 2021), ViT (visual transformer) (Dosovitskiy et al., 2020), nnFormer (Zhou et al., 2021), and nnUNet (Isensee et al., 2021). Both 2D and 3D UNet are FCN-based encoder–decoder structures with convolutional blocks and skip-connections between the encoder and decoder. The UNet++ is a nested-connected encoder–decoder structure, where each convolutional block is connected to all other blocks. The TransUNet is an encoder–decoder network, where the encoder of UNet is replaced by a 2D transformer including a positional embedding scheme followed by a visual transformer (ViT). The nnFormer is a 3D UNet-type framework that

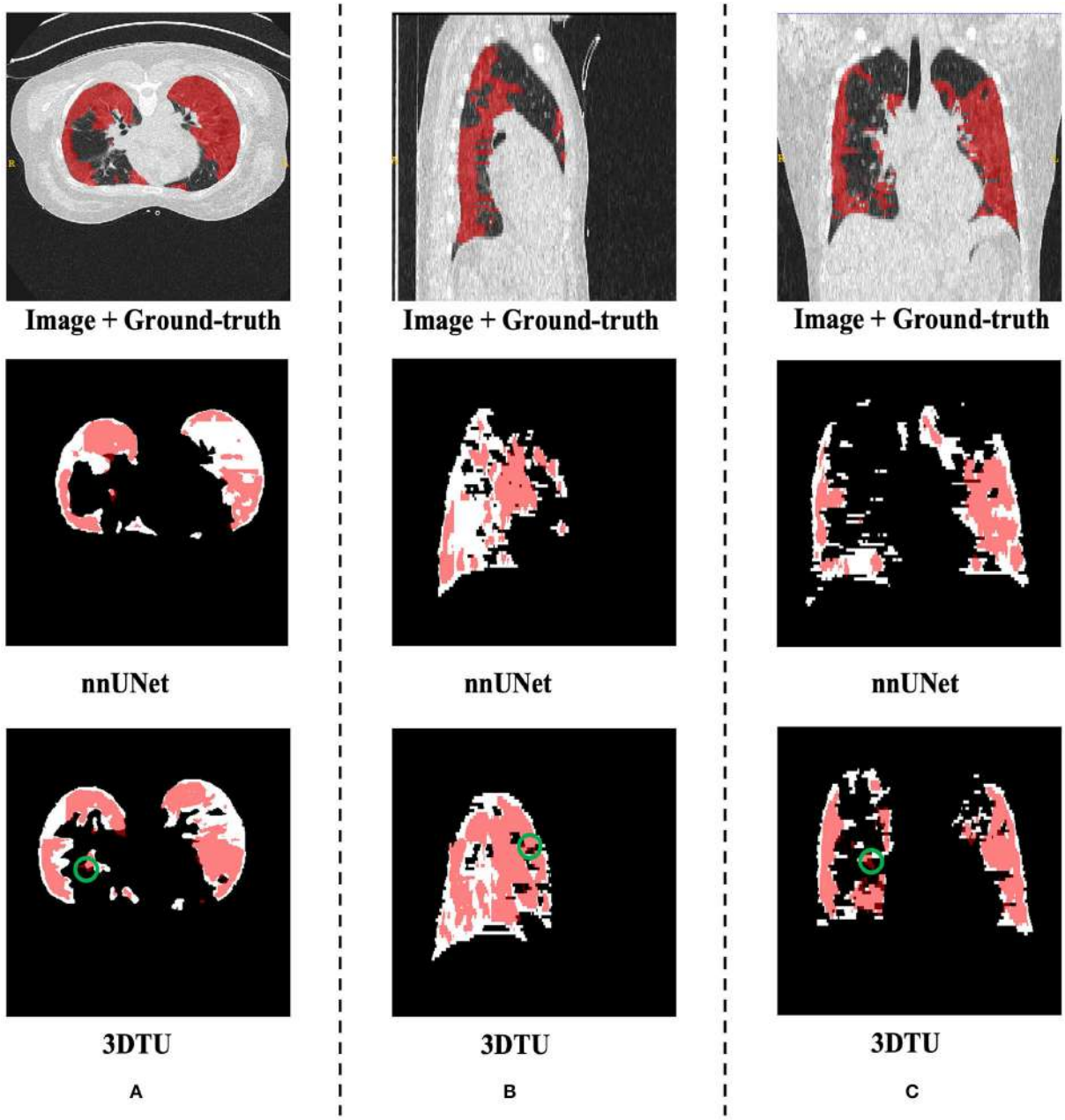


FIGURE 4
Visualization of the infection segmentation results on the Covid20 dataset produced by our 3DTU and nnUNet. Columns **(A–C)** show the *x*–*y* plane, *y*–*z* plane, and *x*–*z* plane of 3D segmentation predictions, respectively. The true-positive regions are highlighted in pink. The false-negative regions are highlighted in red (e.g., the green circle regions in the last row). Better view with colors and zooming in.

replaces the convolutional blocks with three different novel attention mechanisms.

The metrics we used to evaluate our 3DTU include mIoU, DICE score, and Hausdorff Distance (HD). IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between them. For binary (two classes) or multi-class segmentation, the mean IoU (mIoU) of the image is calculated by taking the IoU of each class and averaging them. DICE score is the harmonic mean of precision and recall of the segmentation results. mIOU and DICE scores are two overlap-based metrics measuring the similarity between the ground truths and segmentation predictions. The range of mIOU and DICE scores is from 0 to 1 and the larger value indicates better segmentation performance. The directed

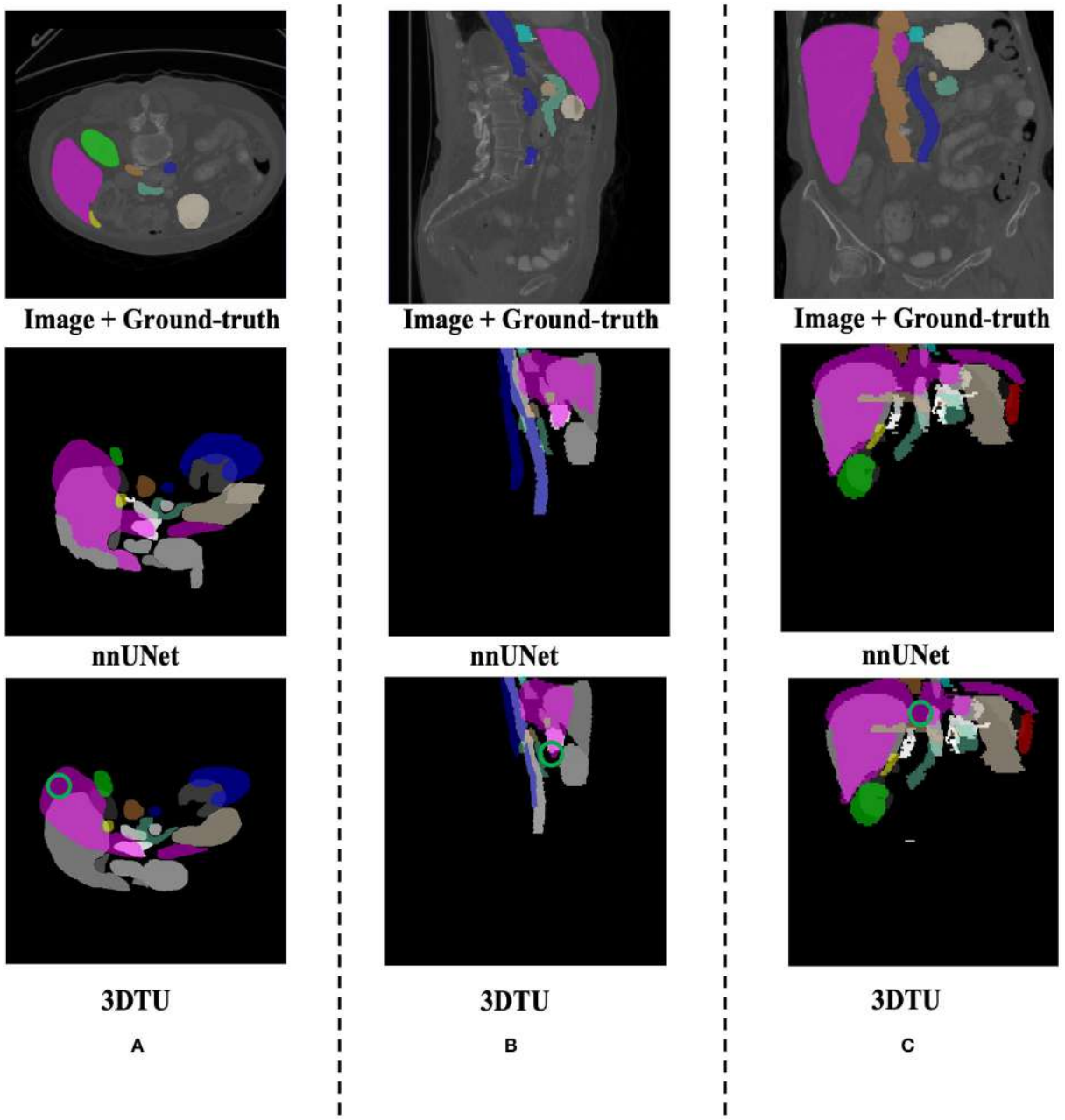


FIGURE 5
Visualization of the segmentation results on the Synapse dataset produced by our 3DTU and nnUNet. Columns **(A–C)** show the *x*–*y* plane, *y*–*z* plane, and *x*–*z* plane of 3D segmentation predictions, respectively. The green circle indicates part of false-negative regions. Better view with colors and zooming in.

average Hausdorff distance (HD) from point set X to Y is computed by the sum of all minimum distances from all points from point set X to Y divided by the number of points in X. HD is a shape distance-based metric, which measures the dissimilarity between the surfaces of the segmentation results and the related ground truths. A lower value of HD indicates better performance.

4.4. Comparative experiments

Table 1 provides the performance of our proposed 3DTU and the six competing baselines, including 2D UNet (Ronneberger et al., 2015), 3D UNet (Ronneberger et al., 2015), UNet++ (Zhou et al., 2019), TransUNet (Chen et al., 2021), visual transformer (ViT) (Dosovitskiy et al., 2020), and

TABLE 2 Dice scores (in %) of our 3DTU on three datasets.

DICE score	Placenta dataset	Covid20 dataset	Synapse dataset
CNN + UNet decoder	68.6	74.3	59.5
BiT + UNet decoder	66.9	72.8	70.2
CNN + BiT	80.0	89.2	65.1
3DTU	**84.0**	**92.0**	**87.3**

The best results are shown in bold.

TABLE 3 Dice scores (in %) of our 3DTU running on data that has been preprocessed with/without positional encoding.

	Placenta dataset	Covid20 dataset	Synapse dataset
3DTU w/o Positional encoding	84.0	92.0	87.3
3DTU with Positional encoding	82.7	92.1	86.8

nnFormer (Zhou et al., 2021) on the Placenta and Covid20 datasets. It shows that our 3DTU outperforms all competing baseline methods consistently in terms of mIOU and DICE scores on both datasets, while beating most of the methods in the baseline in the Synapse dataset, indicating that the segmentation results of our models match well with the ground truth. For example, our proposed 3DTU outperforms baselines with at least 0.48% and 0.44% increases in DICE scores on Placenta and Covid20 datasets, respectively. This may attribute to the attention mechanism proposed in the 3DTU, which can compute the attention scores from three different directions to yield discriminative and anisotropic semantic features for 3D images. In general, the transformer-based methods (e.g., TransUNet, ViT, etc.) perform better than the other baseline methods. In addition, we visualized the segmentation results of our 3DTU and the best baseline method (i.e., nnUNet) on three datasets in Figures 3–5, respectively.

4.5. Ablation study

We conducted an ablation study on both datasets (i.e., Placenta and Covid20) to evaluate the effectiveness of each part in our 3DTU framework. Our 3DTU is an encoder–decoder-based framework, where the encoder consists of a CNN networks part as well as a bi-directional transformer (BiT) part, where the decoder is in the UNet decoder setting. Hence, we designed the following four experiments in our ablation study.

- We removed the CNN networks in the encoder and directly fed the input images to the BiT part.
- We removed the BiT part in the encoder and directly connected the CNN networks to the UNet decoder.
- We removed the UNet decoder part and considered the BiT as both (part of) encoder and decoder.[4]
- We designed a comparative experiment where we trained 3DTU with positional encoded representations. We encoded the representations at the input of the transformer encoder.

The results in Table 2 show the effectiveness and necessity of all the sub-parts in our 3DTU. The results in Table 3 indicate that positional encoding is not necessary in our framework since our attention mechanism can process the 3D data as a whole. Compared with the 3DTU w/o positional encoding, the segmentation dice scores yielded by 3DTU with positional encoding are not changed or even decreased. When we removed the CNN networks and only utilized BiT as the encoder (see results of BiT+Unet decoder in Table 2), the segmentation performance decreased on both datasets (e.g., DICE decrease from 84.0 to 66.9% and from 92.0 to 72.8% on Placenta and COVID datasets, respectively). This indicates an essential role of CNN-based convolutional layers in the encoder, without which the self-attention transformer layers may not localize the raw image pixels precisely. Meanwhile, the segmentation performance increase when we use BiT instead of UNet as a decoder (see results of CNN + UNet Decoder and CNN + BiT). This manifests that, compared with UNet-based methods, the (bi-directional) transformers are more powerful in boosting the segmentation results.

4.6. Parameter analysis

We analyze the impact of two parameters, including the loss weights α and the number of transformer cells, on the segmentation performance of our proposed 3DTU across two datasets in Figure 6. In general, Figure 6 indicates that the segmentation results performed by our 3DTU are consistent. Figure 6A shows that the dice results increase and then decrease with the increase of α from 0 to 1. The best dice scores are achieved when $\alpha = 0.2$ on both Placenta and Covid20 datasets. Figure 6B shows that the segmentation performance improves when increasing the number of transformer cells from 3 to 6. However, the performance will keep stable (on the Placenta dataset) or even slightly decrease (on the Covid20 dataset) when the framework goes deeper. The reason for the slight decrease in the performance of the Covid20 dataset may result from the small size of the dataset. Only 20 3D images are included in the Covid20 dataset, which may not facilitate the training process when the network goes deep. Moreover, our 3DTU has a total of 70M parameters (when training on the Covid20 dataset and the Synapse dataset), which is more than 2D UNet (7M) and

4 It shows in Devlin et al. (2018) that the bi-directional transformer can serve as both encoder and decoder.

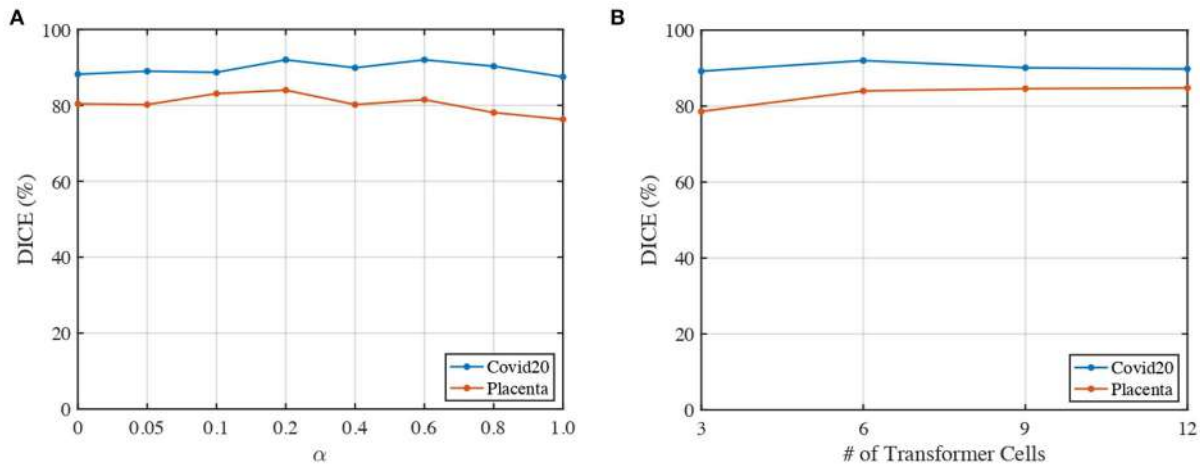


FIGURE 6
Impacts of α and number of transformer cells on segmentation performance. **(A)** Dice of 3DTU vs. α. **(B)** Dice of 3DTU vs. number of transformer cells.

3D UNet (17M) but beats the other transformer-based or hybrid framework in the baseline (the TransUNet has 80M parameters, and nnFormer has 158M parameters).

5. Conclusion

In this article, we propose a novel 3D transformer UNet (3DTU) framework to capture global contextual information for 3D medical image segmentation. A new attention mechanism is proposed with our 3DTU framework, which is especially suitable for computing self-attentions for 3D objects. The experimental results on two 3D medical image datasets demonstrate that our method can outperform several state-of-the-art segmentation baselines. In the future, we plan to explore how to reduce the computation loads in transformer layers, which may improve the efficiency of most current transformer-based methods.

Data availability statement

The Covid20 dataset is from the community of Coronavirus Disease Research-COVID-19 (Jun et al., 2021) and is available *via* https://zenodo.org/record/3757476#.Y1NGmy1h1B1. The Placenta dataset is available upon request.

Author contributions

XF took charge of conception, design, method implementation, statistical analysis, and manuscript writing. ZS and YW took charge of data collection and preprocessing. ZS, EZ, HH, and YW took charge of experimental design, results discussion, and manuscript proofreading. HT and LZ took charge of project design, analysis, interpretation, and manuscript writing/revising. All authors contributed to the article and approved the submitted version.

Funding

This project was partially supported by NSF IIS 2045848 and NIH/NICHD (R01HD094381 and R01HD104822), as well as by the Burroughs Wellcome Fund Preterm Birth Initiative (NGP10119) and the Bill & Melinda Gates Foundation (INV-005417, INV-035476, and INV-037302).

Acknowledgments

We thank the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation (NSF) grant number ACI-1548562 and NSF award number ACI-1445606, which provide the computation resources based on the Pittsburgh Supercomputing Center (PSC) for part of our work. We would like to appreciate the efforts devoted by the community of Coronavirus Disease Research–COVID-19 and Zenodo to collect and share the COVID-19 CT image dataset. Meanwhile, we appreciate the Washington University in Saint Louis for collecting and sharing the data Placenta MRI dataset for our segmentation algorithm evaluations.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern. Anal. Mach. Intell.* 39, 2481–2495. doi: 10.1109/TPAMI.2016.2644615

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., et al. (2021). Swin-Unet: unet-like pure transformer for medical image segmentation. *arXiv preprint* arXiv:2105.05537. doi: 10.48550/arXiv.2105.05537

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., et al. (2021). TransUnet: transformers make strong encoders for medical image segmentation. *arXiv preprint* arXiv:2102.04306. doi: 10.48550/arXiv.2102.04306

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Cham: Springer), 801–818.

Chen, Y., Rohrbach, M., Yan, Z., Shuicheng, Y., Feng, J., and Kalantidis, Y. (2019). "Graph-based global reasoning networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 433–442.

Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., and Sun, J. (2018). "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7103–7112.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: encoder-decoder approaches. *arXiv preprint* arXiv:1409.1259. doi: 10.3115/v1/W14-4012

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Athens: Springer), 424–432.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805

Diakogiannis, F. I., Waldner, F., Caccetta, P., and Wu, C. (2020). Resunet-a: a deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogram. Remote Sens.* 162, 94–114. doi: 10.1016/j.isprsjprs.2020.01.013

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint* arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929

Gao, Y., Zhou, M., and Metaxas, D. N. (2021). "Utnet: a hybrid transformer architecture for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Strasbourg: Springer), 61–71.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). "Generative adversarial nets," *Advances* in *Neural Information Processing Systems 27*. Montreal, QC.

Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H., and Xu, D. (2022a). Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. *arXiv preprint* arXiv:2201.01266. doi: 10.1007/978-3-031-08999-2_22

Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., et al. (2022b)."UNETR: transformers for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (Waikoloa, HI: IEEE), 574–584.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 770–778.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., et al. (2020). "Unet 3+: a full-scale connected unet for medical image segmentation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Barcelona: IEEE), 1055–1059.

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., and Maier-Hein, K. H. (2021). NNU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat. Methods* 18, 203–211. doi: 10.1038/s41592-020-01008-z

Ji, G.-P., Chou, Y.-C., Fan, D.-P., Chen, G., Fu, H., Jha, D., et al. (2021). "Progressively normalized self-attention network for video polyp segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Strasbourg: Springer), 142–152.

Jia, H., Tang, H., Ma, G., Cai, W., Huang, H., Zhan, L., et al. (2021). PSGR: pixel-wise sparse graph reasoning for covid-19 pneumonia segmentation in ct images. *arXiv preprint* arXiv:2108.03809. doi: 10.48550/arXiv.2108.03809

Jun, M., Yixin, W., Xingle, A., Cheng, G., Ziqi, Y., Jianan, C., et al. (2021). Toward data-efficient learning: A benchmark for COVID-19 CT lung and infection segmentation. *Med. Phys.* 48, 1197–1210.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*. Lake Tahoe.

Li, J., Chen, J., Tang, Y., Landman, B. A., and Zhou, S. K. (2022). Transforming medicalimaging with transformers? a comparative review of key properties, current progresses, and future perspectives. *arXiv preprint* arXiv:2206.01136. doi: 10.48550/arXiv.2206.01136

Li, X., Yang, Y., Zhao, Q., Shen, T., Lin, Z., and Liu, H. (2020). "Spatial pyramid based graph reasoning for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle: IEEE), 8950–8959.

Li, Y., and Gupta, A. (2018). "Beyond grids: learning graph representations for visual recognition," in *Advances in Neural Information Processing Systems 31*. Montréal.

Lin, A., Chen, B., Xu, J., Zhang, Z., Lu, G., and Zhang, D. (2022). Ds-transunet: dual swin transformer u-net for medical image segmentation. *IEEE Trans. Instrum. Meas.* 71, 4005615. doi: 10.1109/TIM.2022.3178991

Liu, M., Cheng, D., Wang, K., and Wang, Y. (2018). Multi-modality cascaded convolutional neural networks for alzheimer's disease diagnosis. *Neuroinformatics* 16, 295–308. doi: 10.1007/s12021-018-9370-4

Liu, M., Maiti, P., Thomopoulos, S., Zhu, A., Chai, Y., Kim, H., et al. (2021). "Style transfer using generative adversarial networks for multi-site mri harmonization," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Strasbourg: Springer), 313–322.

Liu, M., Zhu, A., Maiti, P., Thomopoulos, S. I., Gadewar, S., Chai, Y., et al. (2022). Style transfer generative adversarial networks to harmonize multi-site mri to a single reference image to avoid over-correction. *bioRxiv*. doi: 10.1101/2022.09.12.506445

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA: IEEE), 3431–3440.

Noh, H., Hong, S., and Han, B. (2015). "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision* (Santiago: IEEE), 1520–1528.

Pan, X., Zhao, Y., Chen, H., Wei, D., Zhao, C., and Wei, Z. (2020). Fully automated bone age assessment on large-scale hand x-ray dataset. *Int. J. Biomed. Imaging* 2020, 8460493. doi: 10.1155/2020/8460493

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., et al. (2018). "Image transformer," in *International Conference on Machine Learning*. (Stockholmsmässan: PMLR), 4055–4064.

Petit, O., Thome, N., Rambour, C., Themyr, L., Collins, T., and Soler, L. (2021). "U-net transformer: self and cross attention for medical image segmentation," in *International Workshop on Machine Learning in Medical Imaging*. (Strasbourg: Springer), 267–276.

Qiu, Y., Liu, Y., Li, S., and Xu, J. (2021). Miniseg: an extremely minimum network for efficient COVID-19 segmentation. *Proc. AAAI Conf. Artif. Intell.* 35, 4846–4854. doi: 10.1609/aaai.v35i6.16617

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-Net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Munich: Springer), 234–241.

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556. doi: 10.48550/arXiv.1409.1556

Sun, Z., Wu, W., Zhao, P., Wang, Q., Woodard, P., Nelson, D., et al. (2022). Dual-contrast mri reveals intraplacental oxygenation patterns, detects placental abnormalities and fetal brain oxygenation. *Ultrasound Obstetr. Gynecol.* doi: 10.1002/uog.24959

Tokunaga, H., Teramoto, Y., Yoshizawa, A., and Bise, R. (2019). "Adaptive weighting multi-field-of-view cnn for semantic segmentation in pathology," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 12597–12606.

Valanarasu, J. M. J., Oza, P., Hacihaliloglu, I., and Patel, V. M. (2021). "Medical transformer: gated axial-attention for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Strasbourg: Springer), 36–46.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in *Advances in Neural Information Processing Systems 30* (Long Beach Convention & Entertainment Center).

Wang, W., Chen, C., Ding, M., Yu, H., Zha, S., and Li, J. (2021). "Transbts: multimodal brain tumor segmentation using transformer," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Strasbourg: Springer), 109–119.

Xie, Y., Zhang, J., Shen, C., and Xia, Y. (2021). "COTR: efficiently bridging cnn and transformer for 3D medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Strasbourg: Springer), 171–180.

Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. (2017). "Improved variational autoencoders for text modeling using dilated convolutions," in *International Conference on Machine Learning* (Long Beach Convention & Entertainment Center, PMLR), 3881–3890.

Yu, F., and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint* arXiv:1511.07122. doi: 10.48550/arXiv.1511.07122

Yun, B., Wang, Y., Chen, J., Wang, H., Shen, W., and Li, Q. (2021). Spectr: spectral transformer for hyperspectral pathology image segmentation. *arXiv preprint* arXiv:2103.03604. doi: 10.48550/arXiv.2103.03604

Zhang, J., Zhou, L., Wang, L., Liu, M., and Shen, D. (2022). Diffusion kernel attention network for brain disorder classification. *IEEE Trans. Med. Imaging* 41, 2814–2827. doi: 10.1109/TMI.2022.3170701

Zhang, X., Zou, Y., and Shi, W. (2017). "Dilated convolution neural network with leakyrelu for environmental sound classification," in *2017 22nd International Conference on Digital Signal Processing (DSP)* (London: IEEE), 1–5.

Zhou, H.-Y., Guo, J., Zhang, Y., Yu, L., Wang, L., and Yu, Y. (2021). nnFormer: Interleaved transformer for volumetric segmentation. *arXiv preprint* arXiv:2109.03201. doi: 10.48550/arXiv.2109.03201

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., and Liang, J. (2019). Unet++: redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* 39, 1856–1867. doi: 10.1109/TMI.2019.2959609

BMC Medical Imaging

RESEARCH

Open Access

Multimodal Biomedical Image Segmentation using Multi-Dimensional U-Convolutional Neural Network

Saravanan Srinivasan[1], Kirubha Durairaju[2], K. Deeba[3], Sandeep Kumar Mathivanan[4], P. Karthikeyan[5] and Mohd Asif Shah[6,7,8*]

Abstract

Deep learning recently achieved advancement in the segmentation of medical images. In this regard, U-Net is the most predominant deep neural network, and its architecture is the most prevalent in the medical imaging society. Experiments conducted on difficult datasets directed us to the conclusion that the traditional U-Net framework appears to be deficient in certain respects, despite its overall excellence in segmenting multimodal medical images. Therefore, we propose several modifications to the existing cutting-edge U-Net model. The technical approach involves applying a Multi-Dimensional U-Convolutional Neural Network to achieve accurate segmentation of multimodal biomedical images, enhancing precision and comprehensiveness in identifying and analyzing structures across diverse imaging modalities. As a result of the enhancements, we propose a novel framework called Multi-Dimensional U-Convolutional Neural Network (MDU-CNN) as a potential successor to the U-Net framework. On a large set of multimodal medical images, we compared our proposed framework, MDU-CNN, to the classical U-Net. There have been small changes in the case of perfect images, and a huge improvement is obtained in the case of difficult images. We tested our model on five distinct datasets, each of which presented unique challenges, and found that it has obtained a better performance of 1.32%, 5.19%, 4.50%, 10.23% and 0.87%, respectively.

Keywords U-net, Multimodal convolutional neural network, Segmentation, Medical image, MDU-CNN

*Correspondence:
Mohd Asif Shah
drmohdasifshah@kdu.edu.et
[1] Department of Computer Science and Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India, Chennai, India
[2] Department of Computer Science and Engineering, Rajarajeswari College of Engineering, Bangalore 560074, India
[3] School of Computer Science and Applications, REVA University, Bangalore 560064, India
[4] School of Computing Science and Engineering, Galgotias University, Greater Noida 203201, India
[5] Department of Computer Applications,School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India
[6] Department of Economics, Kabridahar University, Po Box 250, Kabridahar, Ethiopia
[7] Centre of Research Impact and Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India
[8] Division of Research and Development, Lovely Professional University, Phagwara, Punjab, 144001, India

Introduction

Segmenting skin lesions using computer assistance becomes difficult due to variations in their shapes and sizes. CAD methods rely on proper lesion segmentation as a crucial early step to obtain precise evaluations of skin lesion borders and sizes. The majority of expert dermatologists have found the subjective clinical segmentation evaluation of skin lesions using deep learning-based techniques to be insufficient [1]. This study primarily aims to utilize advanced techniques, such as deep learning-based automated skin lesion segmentation, to enhance the accuracy and efficiency of melanoma classification, while dermoscopy images aid medical professionals in detecting melanoma in its initial stages. The U-net algorithm, which depends on convolutional neural networks, serves as an indispensable tool for carrying out the segmentation process. Deriving colour and shape features from the segmented image using a local binary pattern along with an edge histogram is highly effective. Various classifiers, such as random forest and naive Bayes, were used to analyze all the features extracted from the skin images to determine if they contained melanoma or benign lesions [2]. This study aims primarily to leverage state-of-the-art approaches, such as automated skin lesion segmentation that uses deep learning techniques, to boost the precision and swiftness of classifying melanomas while also aiding medical professionals in spotting them at an early stage through dermoscopy images. Additionally, the U-net algorithm, which utilizes a convolutional neural network, is a vital component in performing the segmentation. The combination of a local binary pattern and an edge histogram is highly effective for extracting colour and shape features from the segmented image. The analysis of all the extracted features from skin images was conducted by applying different classifiers, including random forest and naive Bayes, to detect the existence of either melanoma or benign lesions [3]. The worldwide fatality rates attributed to melanoma make it one of the deadliest and most aggressive types of skin cancers. Clinical practitioners typically employ biopsy techniques along with microscopic analyses when screening for or preventing skin cancers. The use of a dermatoscope to capture clear images of the affected area is critical in understanding lesion patterns and aiding diagnosis. Additionally, as the lesion evolves and changes its shape over time, manual segmentation of its region can be tough to predict and quite arduous [4].

Melanoma-related deaths due to skin cancer have significantly surged in recent times. However, survival rates may increase for individuals with this chronic condition if melanoma skin disease lesions are detected early. Identifying these lesions is often challenging as they may be obstructed by clinical objects or appear differently due to variations in colour or contrast. Nevertheless, state-of-the-art techniques for sorting and diagnosing involve the use of fully convolutional neural networks that employ an encoder-decoder technique. The encoding layers in these techniques may result in the loss of location information, thus generating granular segmentation-based masks [5]. The reports of rising death toll each year due to melanoma skin cancer indicate a significant increase in its prevalence. Therefore, creating an efficient and successful non-invasive computerized diagnosis tool that precisely detects melanoma by segmenting skin lesions is imperative. This study addresses the segmentation of melanoma skin lesions in dermoscopic images by introducing a new algorithm that utilizes both perceptual colour distinctions and a binary morphological method [6].

The mortality rate for melanoma is high once it has advanced to that stage. However, scientists have worked hard to design automatic systems that can rapidly recognize this fatal sickness. Due to the wide array of skin lesion volumes and shades present in melanoma moles, the identification process is often a challenging task that consumes substantial amounts of time. The presence of noise or blurring, along with luminance modifications in suspicious images, contributes to making detection more complex. To overcome the limitations of previous research, we demonstrate a DL model in this paper [7]. Excessive sunlight exposure can often be attributed as the cause of melanoma, with digital dermoscopy serving as a tool for identifying cancerous areas within skin lesions. To accurately use automated lesion recognition and classification, one must distinguish between healthy skin tissue and cancerous cells. Lesion segmentation has an impact on both the accuracy and precision of classification. Therefore, the focal point of this study is the introduction of a new lesion classification system. Hair filtration, the use of bubbles, and improving specular perception are some of the means to choose from. An improved technique utilizing advanced levelling methodology has been specifically created for the detection and removal of malignant hair [8]. Treating cancerous cells on pigmented skin lesions by simple ablation after visually examining them allows for the timely identification of melanoma. However, performing several biopsies is necessary because visual exams can have variable accuracy levels when there is a shortage of dermatologists who can perform them. To improve the performance of computerized melanoma segmentation methods in dermoscopic images, this work suggests a deep learning-based approach [9]. Melanoma may be lethal, but if detected at an early stage, it can be treated and cured. However, accurate identification of squamous cell carcinoma versus a benign tumor is essential. For this reason, the use of computer-generated recognition to obtain dermoscopy

images has become popular, and automated and accurate classification of melanoma is the primary goal of this study. Analyzing the efficiency of this approach in terms of identifying melanoma is done using the ISBI 2016 skin lesion assessment dataset [10].

Related work

Shifa Kubra N et al. [11], the researchers explored the capability of deep convolutional neural networks in distinguishing between benign and cancerous skin cells. Their study utilized a sample size of 3600, with 3000 samples used for training and the remaining portion used for validation, specifically focusing on dermoscopy images. The findings indicated that deep learning models outperformed human dermatologists in terms of accuracy. By employing very deep neural networks along with switch reversal and fine-tuning on dermoscopy images, the researchers were able to achieve superior diagnostic capabilities compared to experienced clinicians and oncologists. In another study by Chen Zhao et al. [12], the researchers emphasized the significance of fully connected neural networks and U-Net in the melanoma segmentation process. However, they identified that as the depth of these deep neural networks increased, the issue of gradient vanishing arose, making them susceptible to parametric redundant systems. This issue had a negative impact on skin lesion image segmentation and resulted in a decrease in the Jaccard index value. To address these challenges and improve the survival rate of melanoma patients, the paper proposed an enhanced segmentation process based on U-Net++. This approach aimed to tackle the mentioned issues and enhance the accuracy of melanoma segmentation. Andrea Pennisi et al. [13] aimed to create an AI-based system that assesses relationships between images taken at different points in time. The first step in this process is to divide up the affected part of the lesion. Moreover, to detect the edges of skin lesions in medical images, we propose using an attention squeeze U-net model based on deep learning techniques. Based on the quantitative outcomes obtained from an open-access dataset, it appears feasible to achieve accurate segmentation using a simplified approach.

Nojus Dimša et al. [14], encoder and decoder structures are beneficial for object segmentation, particularly the U-Net framework, which serves as the foundation for segmenting medical images in a network system. Various combinations of U-Net-type layouts have been introduced recently in an effort to improve segmentation outcomes. Thus, we evaluated the ability and effectiveness of three U-Net type models, namely U-Net, U-Net++, and MultiResU-Net, for the multi-class segmentation of melanoma. Lina Liu et al. [15] propose utilizing an advanced deep convolutional neural network framework referred to as the U-Net model to perform accurate segmentation of skin lesions. By introducing batch normalization layers in our modified version of U-Net, along with an enhanced convolutional neural network architecture, we were able to prevent prediction errors and enrich the perceptron during the training phase. Results of experimental evaluation have demonstrated that adding enlarged convolution can considerably enhance the effectiveness of the presented method. Additionally, we present a simple, direct, yet practical experimental ensemble approach that does not require training additional frameworks.

Haoran Lu et al. [16] demonstrate in this research that the success of U-Net in the field of healthcare image classification is primarily attributed to the partitioning solution it employs, rather than the merging of different features. Half-U-Net is introduced as a result of this observation because it primarily facilitates the joint optimization part of the process. Unifying the channel numbers, employing feature-length fusion, and making use of Ghost modules are the three primary methods by which Half-U-Net reduces the complexity of the network. According to experimental findings, the proposed Half-U-Net achieved superior segmentation performance and reduced the network's complexity when compared to other U-Nets and their different versions. Omran Salih et al. [17] developed this article by using a binarization convolution operation instead of a standard convolution layer, creating a local binarization convolutional neural network (LBCDN) for comprehensive skin lesion segmentation, which greatly improved accuracy. The LBCDN architecture was proposed to minimize computational cost by combining an improved deep neural network with a smaller encoder and decoder system. The LBCDN framework achieved the highest dice coefficient among all other methods, demonstrating its superior and stable performance. Yadi Zhen et al. [18] highlight the challenges of defining the perimeter of melanoma due to its irregular shape, structure, and colour. In this paper, a better DC-U-Net network-based segmentation algorithm is developed to address these issues. To sharpen the model's focus on the melanoma lesion area, a connection-focused ECA-NET module has been added. In order to further refine the segmentation results, conditional random field and test data augmentation are used as post-processing techniques.

Zahraa E. Diame et al. [19] evaluated five frameworks (U-Net, Res-U-Net, VGG-16UNET, DenseNet-121, and EfficientNet-B0) to assess the potential application of deep learning techniques for skin lesion segmentation to identify lesion boundaries. The DenseNet-121 framework outperformed other methods in terms of precision rate across all training datasets. Prashant Brahmbhatt et al.

[20] focus on segmenting the well-known skin lesion problem using an ensemble method. The paper combines the conventional strategy and the ensemble principle to achieve acceptable performance. The secondary goal is to reduce the time spent on image pre- and post-processing. The PH2 dataset consists of dermoscopic images of skin conditions and their corresponding ground truths, obtained through the traditional manual method.

Yangling Ma et al. [21] recommend a multi-instance, learning-based, end-to-end approach for melanoma identification and lesion segmentation simultaneously. They utilize multi-instance content based on a graph-convolutional neural network to recognize melanoma and fully leverage the information in high-resolution photos. The end-to-end approach treats segmentation and identification as inherently connected methods, where a high Jaccard index also indicates the stability of melanoma identification. Nawaz M et al. [7] present a deep learning approach to overcome the limitations of previous work. After completing the pre-processing procedure, they utilize the Corner-Net framework, an image detection technique, to diagnose melanoma lesions. The regional moles are then subjected to the fuzzy K-means clustering method for semantic segmentation. The proposed strategy is evaluated using two standard databases, ISIC-17 and ISIC-18, to assess its segmentation ability. Multiple tests have been conducted to demonstrate the reliability of the proposed strategy, using both numerical metrics and visual representations. Baiju Babu Vimala et al. (2023) [22] employ a hybrid deep learning approach to clean up breast ultrasound images with local speckle noise. Initially, logarithm and exponential modifications are applied to improve the brightness of ultrasonography breast images, followed by the use of guided filter techniques to enhance the detail of proliferative ultrasonography images.

Saravanan et al. [23] sparse coding estimates is utilised for higher-dimensional data, and the encoding scheme employed is metadata-based vector encoding. The atoms of nearby limitation are constructed on the basis of a well-organized k-neighboured system, which preserves the geometric structure of the supervised data. Saravanan et al. [24] kirsch's edge detectors detect boundary edge pixels, which are contrast adaptive histogram equalised. This augmented brain image is then Ridgelet transformed to obtain Ridgelet texture feature parameters. Features are obtained from Ridgelet transformed coefficients, enhanced using Principal Component Analysis, and categorized into Glioma or non-Glioma brain pictures using Co-Active Adaptive Neuro Fuzzy Expert System classification. Shivangi et al. [25] an exhaustive empirical assessment of convolutional neural networks (CNNs) applied to large-scale image classification of gait signals converted into spectrogram images and deep dense artificial neural networks (ANNs) employed for voice recordings has showcased remarkable superiority over existing state-of-the-art methods in disease prediction. The VGFR Spectrogram Detector achieved an impressive classification accuracy of 88.1%, while the Voice Impairment Classifier attained a remarkable 89.15% accuracy. Saravanan et al. [26] leveraging the automated feature extraction capabilities of a three-dimensional deep convolutional autoencoder (3D-DCAE), a novel method has been developed that integrates a neural network-based classifier to construct a unified framework capable of supervised training, achieving the pinnacle of classification accuracy for both ictal and interictal brain state signals. To thoroughly evaluate our method, two distinct models were meticulously crafted and assessed, employing three separate EEG data section lengths and a rigorous tenfold cross-validation procedure.

U-shaped encoder-decoder network architecture

In order to accomplish semantic segmentation, the U-Net employs a fully convolutional network, which is similar to the semantic segmentation method and a fully convolutional network method. The network is constructed to be symmetrical, with an encoder that detects spatial features in the image, and a decoder that uses those features to build a segmentation map. The encoder utilizes a conventional convolutional network architecture. It begins with a pair of 3×3 convolutions, followed by a max pooling function with a pooling dimension of 2×2 and a stride of 2. This process is repeated four times, doubling the number of convolution layer filters after each downsampling. The encoder is connected to the decoder through a series of two 3×3 convolutional operations. On the other hand, the decoder first up-samples the feature map using a 2×2 transposed convolutional operation, resulting in a reduction in the number of feature channels by half. This is followed by a sequence of two 3×3 convolutional operations. This upsampling and two-convolutional processing cycle is repeated four times, with the number of filters in each cycle halved to match the encoder. The final segmentation map is created using a 1×1 convolutional technique. The Rectified Linear Unit (ReLU) activation function is used by all layers except the last convolutional layer, where the sigmoid activation function is employed. One of the most distinctive features of the U-Net design is the use of skip connections. The output of the convolutional layer is passed on to the corresponding layer in the decoder before the pooling process of the encoder. Figure 1 depicts the architecture of U-Net, incorporating an encoding path and a decoding path with skip connections between the appropriate layers.

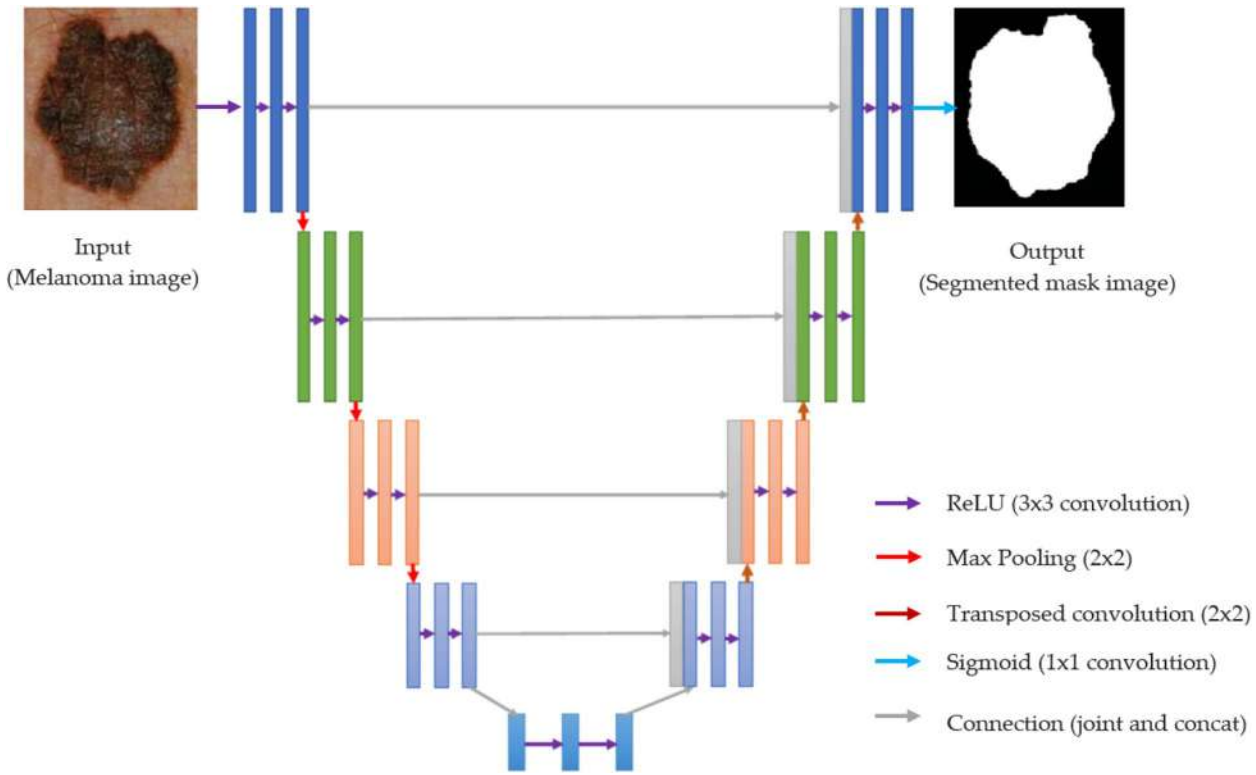


Fig. 1 Architecture of U-shaped encoder-decoder network

The up-sampling procedure results in concatenated feature maps, which are then propagated to the subsequent layers. This allows the network to recover spatial features that may have been lost during pooling operations, thanks to the skip connections. In the context of parametric segmentation, the U-Net design was modified to become a three-dimensional U-Net. This involved replacing the 2D convolution max-pooling and asymmetrical operations with their 3D counterparts. To reduce the number of variables, however, the depth of the network was reduced by one, and the number of filters was doubled before the pooling layers to avoid bottlenecks. In the original U-Net, batch normalization was not used. However, batch normalization was tested in the 3D U-Net, and surprisingly, the results indicated that batch normalization could sometimes decrease performance.

Modification of scale factor - medical images

In medical image analysis, our goal is to separate nuclei, organ systems, and carcinomas from image data captured by different types of equipment. However, these objects of interest often exhibit non-uniform and

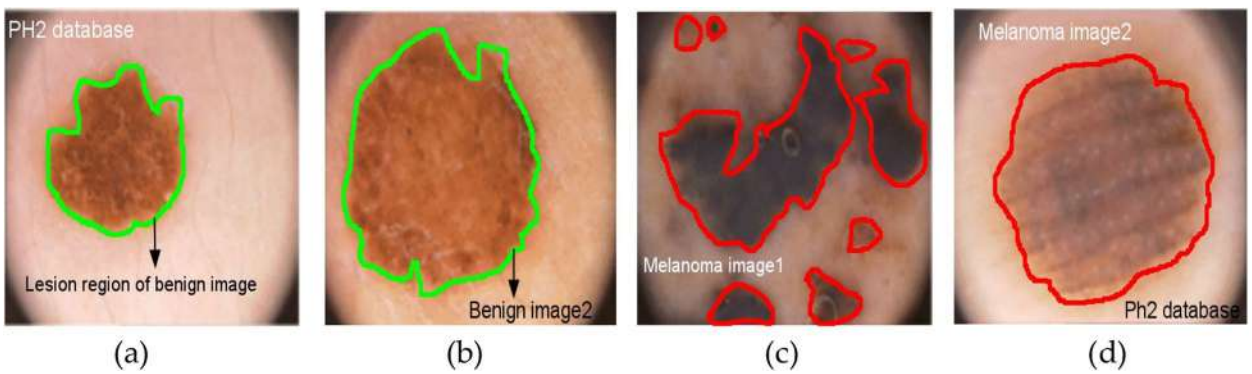


Fig. 2 Sample lesion-dermoscopic images from PH^2 database; **a**, **b** sample images of benign type; **c**, **d** sample images of melanoma type

varying scales. For example, as shown in Fig. 2, dermoscopy images can vary greatly in the size of skin lesions. These sample lesion dermoscopy images are obtained from the PH2 database [27]. Such variations are commonly encountered in various medical image segmentation tasks. Therefore, a network's ability to handle such variations becomes crucial in performing analyses on different entities at different levels. While several deep vision-based studies have addressed this issue, it has not been adequately addressed in the field of medical semantic segmentation. The U-Net framework utilizes a series of two 3×3 convolution operations after every pooling layer and transposed convolution layers. This two-step convolution process is similar to a 5×5 convolution procedure. To enhance U-Net with multi-dimensional analysis, an effective approach is to incorporate 3×3 and 7×7 convolutional processes alongside the 5×5 convolution layer. By replacing the convolution layers with blocks similar to Inception, the U-Net architecture can better balance the learned characteristics from the image at different sizes. Although depth-wise convolutions could be considered, our testing phase showed that incorporating Inception-like blocks outperformed this approach. Despite the performance improvement, it's important to note that including more convolution layers in parallel may increase the memory requirements.

We utilize a series of compact 3v3 convolution blocks to factorize the larger and more challenging 5×5 and 7×7 convolutional layers. The second and third 3×3 convolution blocks yield results that closely resemble those of 5×5 and 7×7 convolutions, respectively. To capture spatial characteristics at various scales, we collect the outputs from these three convolution blocks and concatenate them together. Our research findings indicate that the results from this condensed block are nearly identical to those from the memory-intensive Inception-like structure. This supports the hypothesis that adjacent layers in a visual network are correlated, and this modification significantly reduces the required memory. The memory impact is primarily due to the quadratic increase in the number of filters in the first convolutional layer in deeper network models. To avoid excessive memory utilization, instead of maintaining an equal number of filters in each of the three successive convolutional layers, we gradually increase the number of filters in specific layers (from one to three) to prevent the memory load from the previous layer from propagating excessively into the core part of the network. This allows us to capture spatial attributes derived from distinct context sizes. However, we do not use 3×3, 5×5, and 7×7 filters simultaneously; rather, we interpolate the larger and more expensive 5x5 and 7×7 filters as a series of 3×3 filters (as shown in Fig. 4). Additionally, we incorporate a residual network, known for its effective segmentation process in biomedical images, and the inclusion of 1×1 convolution layers, which provide additional spatial information interpretation. Both of these factors contribute to the success of our approach (as shown in Fig. 3). The multi-dimensional block demonstrates how we progressively increase the number of filters in the three layers following it, along with the inclusion of a residual network (and a 1×1 filter to maintain dimensions).

Possible conceptual difference between encoder-decoder levels

An intensive U-Net model contribution was the prime reason for making shortcut links between the respective layers, and it is placed prior and after the max-pooling, de-convolutional layers, respectively. It lets the network send from encoder to decoder the spatial details that may

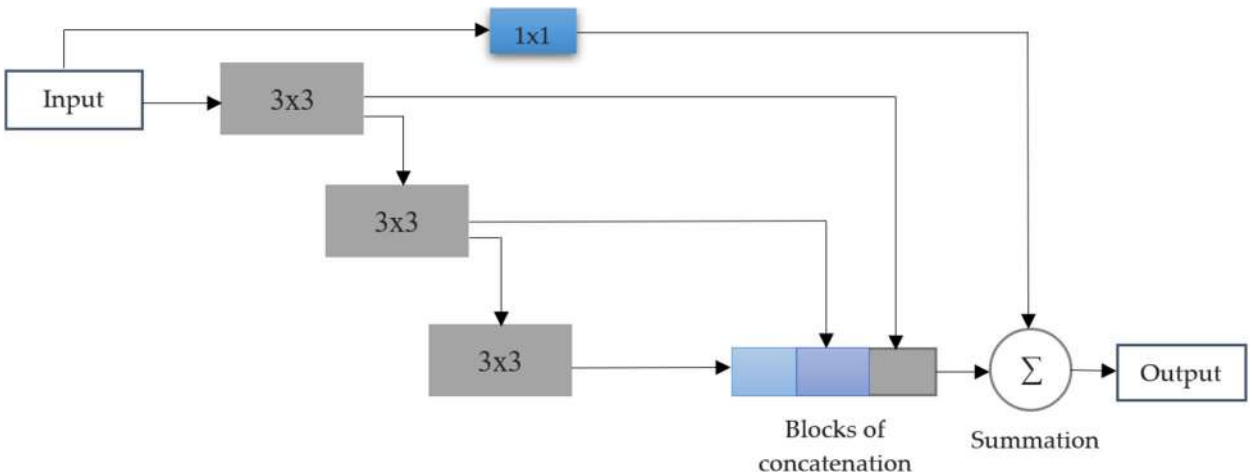


Fig. 3 Multi-dimensional block

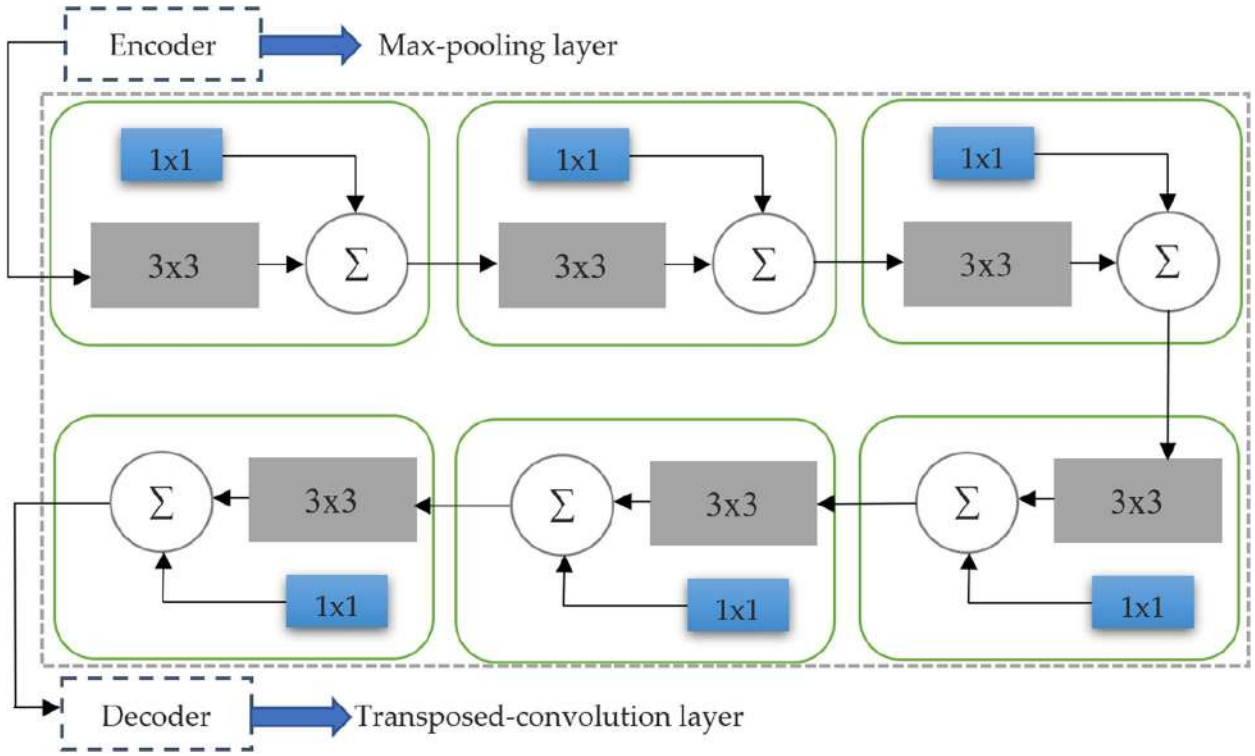


Fig. 4 Proposed encoder-decoder convolution path

have been lost during the pooling process. Maintaining dispersed spatial features, a shortcut link establishes a connection between the encoder that comes prior to the first pooling and the decoder that comes after the final deconvolution process.

When considering the assumptions regarding encoder feature inputs in neural networks, it has been observed that these inputs originate from lower-level computations in the network. Conversely, the features in the decoders are computed in deeper levels, resulting in higher-quality representations. However, performing this operation effectively may require a significant amount of available processing resources. Nevertheless, it is recognized that there may be a conceptual discrepancy when combining these distinct feature sets, which could potentially affect our prediction method during the learning stage. However, as we progress through the shortcut connections, the level of difference is expected to diminish considerably. This can be attributed to the integration of encoder and decoder functions, working together to achieve intensive processing. To address the feature gap between the encoder and decoder, we suggest integrating additional convolutional layers accompanied by shortcut links. This helps to reduce the disparities and compensate for any additional processing performed during decoding through nonlinear transformations on the propagated features. Additionally, we enhance the standard convolutional layers by incorporating residual connections, which have demonstrated significant capability in representing medical images. This concept draws inspiration from image-to-image conversion tasks performed using convolutional neural networks, where the use of pooling layers often results in data loss. Instead of simply appending the feature maps from the encoder to the decoder, we employ a series of convolutional layers with residual blocks, concatenating them with the features from the decoder stage. In Fig. 4, we demonstrate that the combination of encoder and decoder feature maps involves running the encoder features through several convolutional layers first. This is expected to narrow the conceptual differences in feature maps between the encoder and decoder through the application of nonlinear procedures. Additionally, the implementation of residual connections facilitates faster learning and is highly beneficial in deep neural networks.

Proposed MultiDimensional U-CNN architecture

MultiDimensional U-CNN is a promising technology with the potential to revolutionize various medical fields. Its ability to accurately segment complex anatomical structures and pathological lesions can empower clinicians to make better-informed decisions, ultimately

leading to improved patient outcomes. We replace the sequence of two convolutional layers in the MultiDimensionalU-CNN model with the proposed MultiDimensional block, which was discussed in "Proposed MultiDimensional U-CNN Architecture" section. We assign a parameter called M to each of the multidimensional blocks. This parameter governs the number of filters that are used in the convolutional layers that are contained within that block. We calculate M to ensure that the number of parameters in the original U-Net and those in the proposed model have a similar relationship.

$$M = \beta \times F_n \tag{1}$$

β—> scaler co-efficient, F_n—> number of U-Net layer filters.

Decreasing M into F_n and β is a practical way to cut down on the number of parameters while keeping them the same as in U-Net. Our recommended model is compared to a U-Net with filter sizes of [16; 32; 64; 128; 256; and 512] along the stages, and these values are similar to our F_n. Since we wanted to keep the number of model parameters lower than the U-Net, we opted $\beta = 1.56$. We emphasised the importance of gradually increasing the number of filters in the subsequent convolutional layers within a multidimensional block rather than maintaining the same number. Assigning filters $\left[\frac{M}{6}\right]$, $\left[\frac{M}{3}\right]$ and $\left[\frac{M}{2}\right]$ to the three convolutional layers in order yielded the best results in our research. Also, M is doubled after every pooling and deconvolution operation, just like the U-Net. In addition to the MultiDimensional blocks, the proposed encoder-decoder convolution path would replace the normal shortcut connections. For this reason, we perform a few convolutional processes on the extracted features that are being sent from the encoder to the decoder. By getting closer to the internal shortcuts, we expect the semantic difference between the decoder and encoder feature maps to be less noticeable. All along the encoder-decoder convolutional path, we also employ a decreasing number of convolutional blocks. The Rectified Linear Unit (ReLU) activation function is utilized in this network for all of the convolutional layer activation, with the exception of the output layer. All the convolutional layer activation in this network is batch normalized. The output layer, such as the U-Net model, is activated by a sigmoid activation function. Figure 5, depicts the architecture of proposed MultiDimensionalU-CNN architecture. The proposed multidimensional block replaces the sequences of two convolution layers in U-Net frameworks. In addition, rather than using simple shortcut connections, we employ the proposed

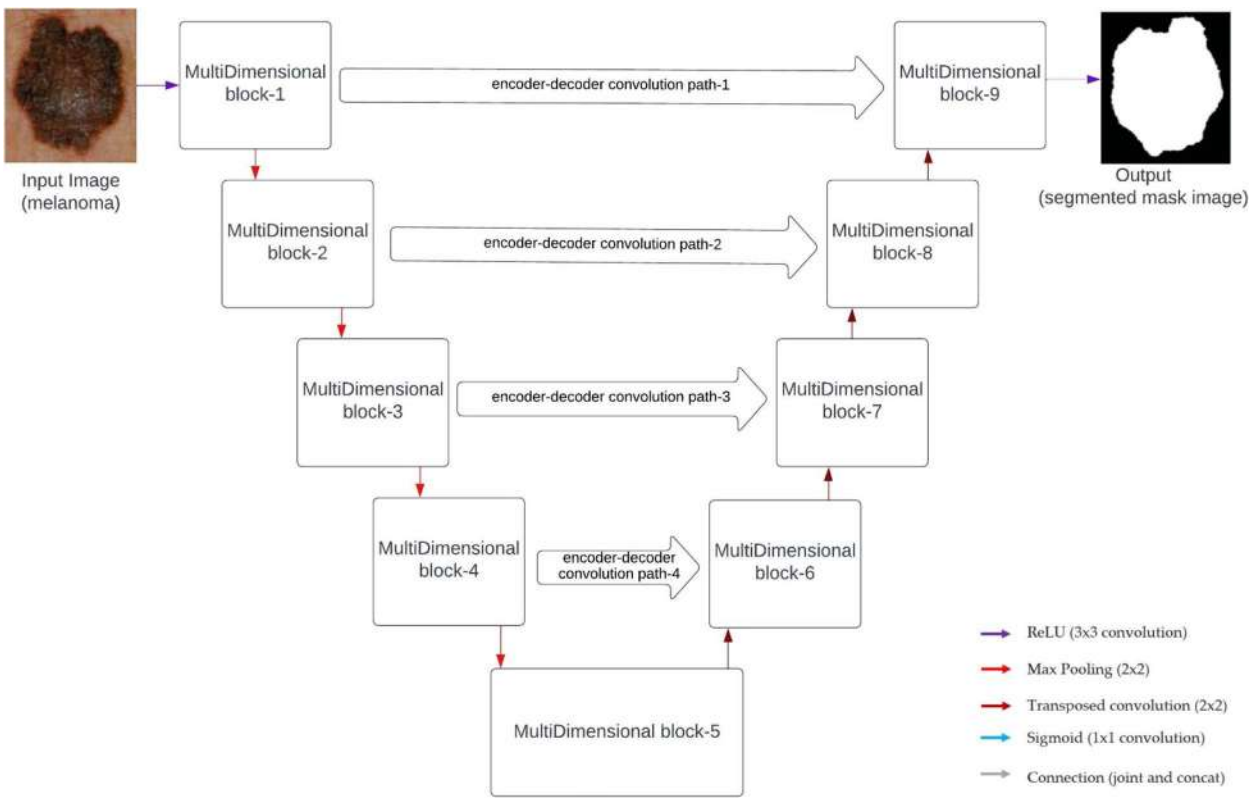


Fig. 5 Proposed MultiDimensionalU-CNN architecture

Table 1 Detailed architecture of MultiDimensionalU-CNN

MultiDimensionalU-CNN					
Block Name (B)	**Size of the Filter**	**No. of filters**	**Convolution Path**	**Size of the Filter**	**No. of filters**
MultiDimensional B1	2D-conv(3,3)	7	encoder-decoder convolution path-1	2D-conv(3,3)	16
	2D-conv(3,3)	18		2D-conv(3,3)	16
MultiDimensional B9	2D-conv(3,3)	26		2D-conv(3,3)	32
	2D-conv(1,1)	52		2D-conv(1,1)	32
MultiDimensional B2	2D-conv(3,3)	18		2D-conv(3,3)	32
	2D-conv(3,3)	36		2D-conv(3,3)	32
MultiDimensional B8	2D-conv(3,3)	54		2D-conv(3,3)	32
	2D-conv(1,1)	106		2D-conv(1,1)	32
MultiDimensional B3	2D-conv(3,3)	36	encoder-decoder convolution path-2	2D-conv(3,3)	64
	2D-conv(3,3)	73		2D-conv(3,3)	64
MultiDimensional B7	2D-conv(3,3)	107		2D-conv(3,3)	64
	2D-conv(1,1)	214		2D-conv(1,1)	64
MultiDimensional B4	2D-conv(3,3)	72		2D-conv(3,3)	64
	2D-conv(3,3)	144		2D-conv(3,3)	64
MultiDimensional B6	2D-conv(3,3)	216	encoder-decoder convolution path-3	2D-conv(3,3)	128
	2D-conv(1,1)	428		2D-conv(1,1)	128
MultiDimensional B5	2D-conv(3,3)	146		2D-conv(3,3)	128
	2D-conv(3,3)	292		2D-conv(3,3)	128
	2D-conv(3,3)	428	encoder-decoder convolution path-4	2D-conv(3,3)	256
	2D-conv(1,1)	856		2D-conv(1,1)	256

encoder-decoder convolution paths. Table 1 represents the detailed description of MDU-CNN architecture.

Experimental setup-dataset

Medical imaging dataset curation is more difficult than conventional computer-aided dataset curation. The high cost of imaging equipment, the complexity of image acquisition pipelines, the need for expert interpretation, and concerns about privacy make it hard to make medical imaging datasets. To this end, only a limited number of publicly available benchmark datasets for medical imaging exist, each of which contains only a few examples of diagnostic images. We wanted to examine the performance of proposed framework by putting it through a number of different imaging techniques. In particular, we chose datasets that were as different as possible from one another. Table 2 contains the detailed description of various datasets utilized in the proposed system.

Magnetic resonance image

All kinds of previously mentioned datasets include 2D clinical images, then we used MRI images from the BraTS2020 database in order to examine the feasibility of our proposed framework for 3D medical imaging. This dataset contains 210 glioblastoma (HGG) and

Table 2 Detailed description of various dataset

Methods	**No. of dataset**	**Total number of images**	**Image resolution**	**Required resolution (input)**
Magnetic resonance image	BraTS2020	230 high-grade, and 85 low-grade gliomas	240×240×155	80×80×48
Non-invasive dermoscopy	ISIC2020	3213	not consistent	256×192
Microbes Fluorescence Microscopy	Murphy Lab 3D hela	106	not consistent	256×256
Endoscopy	EndoSLAM	714	384×288	256×192
Electron Microscopy	ISBI-2022	45	512×512	256×256

75 lower-grade glioma (LGG) multimodal MRI scans [28]. These multimodal scans include native (T1), post-contrast T1-weighted (T1Gd), T2-weighted (T2) and T2 Fluid Attenuated Inversion Recovery (FLAIR) volumes, which were acquired following different clinical protocols and various scanners. The images are of dimensions 240 × 240 × 155 but have been resized to 80 × 80 × 48 for computational ease. All the four modalities, namely, T1, T1Gd, T2 and FLAIR are used as four different channels in evaluating the 3D variant of our model.

Non-invasive dermoscopy

The images of dermoscopy were obtained from the ISIC2020 database (Lesion Boundary Segmentation Task). The ISIC2019 dataset and the HAM-10000 dataset were used to get the data for this task [29]. There are a total of 3213 images of various skin lesions, all of which have expert annotations. The images were originally at a wide variety of resolutions but were scaled down to 256×256 while preserving their median aspect ratio for fast computation.

Microbes fluorescence microscopy

The Murphy Lab's dataset of microbe's fluorescence microscopic analysis was used for this research. The 106 fluorescent microscope image dataset contains a maximum of 4106 cells. There are 50 percent U2OS cells and 50 percent NIH3T3 cells, and the nuclei were manually segmented by specialists [30]. This dataset of the bright-field microscopy images is difficult to analyse because of the inconsistent brightness of the nuclei and the presence of visible debris. Due to computational constraints, the actual size of the image's ranges from 1349×1030 - 1344×1024, but they have been scaled to 256×256.

Endoscopy

The EndoSLAM, a gastrointestinal image database, was utilised for our endoscopy image investigations, and this dataset contains images extracted from 38 video sequences of colonoscopy. However, the images that contained polyps were taken into consideration, resulting in the collection of 714 images [31]. The images had a resolution of 386×292 when they were first captured, after it has been downsized to 256×256 while keeping their aspect ratio.

Electron microscopy

We used the dataset from the ISBI2022: 2-dimensional EM segmentation task to analyse the efficacy of the framework with electron microscope images. A total of 45 images from ssTEM (transmission electron-microscopy) of the ventral nerve cord of Drosophila first instar larvae are included in this collection [32]. The images originally had a resolution of 640×480 pixels, but because of computational constraints, they were downsized to 256×256 pixels.

Table 3 Different dimensional method and parameters used in proposed method

Two-dimensional		Three-dimensional	
Type	**Parameters**	**Type**	**Parameters**
Traditional U-Net	7,961,534	Traditional U-Net	19,565,211
Proposed framework	7,162,381	Proposed framework	17,981,297

Experimental analysis and baseline model

Python's programming language known as Pyhton3 was utilized in conducting our study. By utilizing Keras and its Tensor Flow backend we were able to construct the network models. To execute the tests have been used on a laptop computer that is integrated with features such as Intel Core i5-7700 Processor having frequency up to 4.4 GHz and also comprised with enough Memory space up to 16 GB RAM along-with-Graphics which is powered by MSI Nvidia RTX 3060 having enough strength which is around 12 GB GDDR6, to identify the most effective segmentation approach for clinical images we compared how well MultiDimensionalU-CNN performed relative to U-net. To maintain consistency in the number of parameters between our proposed MultiDimensionalU-CNN model and Classic U-net model we employed a six-layered deep encoder-decoder configuration on latter ensuring filters range from as low as 16 through till high-end i.e., 512. Building a 3D version of MDU-CNN can be easily done by just replacing all its core components with their appropriate three-dimensional counterparts, and there aren't any extra changes or additions that take place during this phase of construction. The different dimensional methods and its respective parameter utilization by the proposed and traditional U-Net model information is shown in Table 3 and Fig. 6.

Image resizing and training mode

The goal of the experiment analysis is to examine the proposed MDU-CNN architecture is better than the original U-Net model. As a result, there was no pre-processing done that was specific to the domain. Input images were just downscaled so they would fit in the GPU-memory, and their image pixels were partitioned by 255 so they would fall in the [0:::1] range before being processed. Similarly, there was no post-processing done that was specific to the domain as well. Because the activation function of the last layer is a Sigmoid, its output values are also in the range [0:::1]. Hence, we used a 0.5 threshold to produce the input image segmentation map. The goal of semantic segmentation is to determine, for each individual pixel,

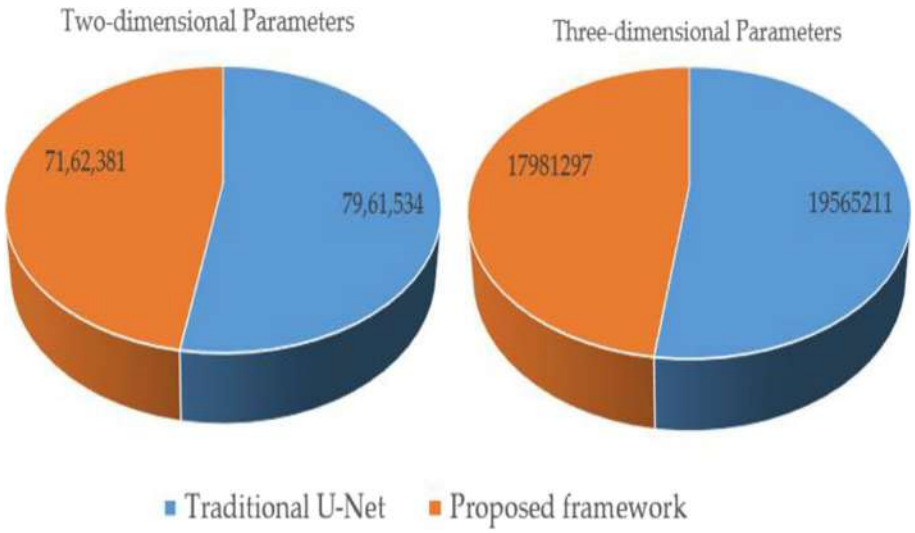


Fig. 6 Proposed and traditional framework parameter utilization demonstration

whether it is a potential point of interest or simply a part of the background. As a result, we may simplify this issue to a binary classification problem on a pixel-by-pixel basis. As a result, in order to calculate the loss function for the network, all we did was minimize the binary cross-entropy factor. Let, A is an image, and B is a segmentation mask of ground truth, it can be computed by the given model is $\widehat{B}$. Here, for pixel $p_i a$, then the network computes $\widehat{b}_{p_i a}$, likewise the $b_{p_i a}$ is the value of the ground truth. The definition of that image's binary cross-entropy loss is as follows:

$$Cros\ Entropy\left(A,B,\widehat{B}\right)=\sum\nolimits_{p_i a\in A}-(b_{p_i a}\log\left(\widehat{b}_{p_i a}\right)+\left(1-b_{p_i a}\right)\log(1-\widehat{b}_{p_i a})) \tag{2}$$

When, the batch is having 'm' images, then the function of loss (I) can be written as,

$$I=\frac{1}{m}\sum\nolimits_{i=1}^{m}Cross\ Entropy\left(A_i,B_i,\widehat{B}_i\right) \tag{3}$$

We tried to reduce the loss of binary cross-entropy as much as possible, hence we trained the model via stochastic gradient descent optimizer. Estimates of the initial and second values of the gradients are used in stochastic gradient computation of diverse learning levels appropriate for each of the parameters. The stochastic gradient is equipped with a set of parameters, two of which, $\alpha 1$ and $\alpha 2$, are responsible for controlling the rate of decay of the first and second moments, respectively. To train the models, a stochastic gradient optimizer was used for a total of 150 iterations. The number of iterations was set at 150 since, after that point, there was no apparent progress in either model.

Performance metric evaluation

During the semantic segmentation process, the places of interest usually make up a small part of the whole image. Metrics like precision and recall are insufficient and frequently lead to a misleading perception of superiority, which is augmented by the faultless detection of the background. As a result, the Jaccard Index has found widespread application in the process of evaluating and benchmarking various segmentation methods and object localization techniques. The Jaccard index (JI) is defined with two pairs, X and Y, as the proportion of the set's intersection and association. This ratio can also be written as,

$$JI=\frac{intersection}{association}=\frac{X\cap Y}{X\cup Y} \tag{4}$$

In this particular scenario, set X denotes the binary-segmentation mask B that correlates to the ground truth, and set Y relates to the binary segmentation mask $\widehat{B}$ that was predicted. As a result of using the JI as the measure, it is not only emphasising accurate segmentation, but it also penalises both under and over segmentation.

Table 4 Comparison outcomes of proposed and traditional method validation accuracy based on k-fold cross-validation

Type	Proposed MDU-CNN (%)	Traditional U-Net (%)	Comparative Increase in performance (%)
Magnetic Resonance Image	79.2130 ± 0.8182	77.2289 ± 0.6923	1.3241
Non-invasive dermoscopy	81.3188 ± 0.4423	76.1256 ± 3.9834	5.1932
Microbes Fluorescence Microscopy	92.6228 ± 0.9816	88.1209 ± 1.9923	4.5019
Endoscopy	83.1567 ± 1.6822	72.9190 ± 1.3989	10.2377
Electron Microscopy	88.8651 ± 0.8012	87.9929 ± 0.7717	0.8722

K-fold cross validation

Leave-one-out cross-validation methods are used to test an algorithm's overall performance on a dataset that is not dependent on it. These tests keep a proper balance between variability and bias. The dataset C is subclassified in to k mutually unique subsets $C_1, C_2, C_3, \ldots . C_k$ for the k-fold cross-validation test. Each iteration of the algorithm uses one of the k divides as the testing dataset and the remainder as the learning set. To examine the segmentation performance of the traditional U-Net and the proposed MultiDimensionalU-CNN framework, Leave-one-out cross-validation tests were carried out on each of the distinct datasets. However, it is a supervised neural pipeline, and the highest performing outcome on the testing set was attained over the maximum quantity of epochs (150) that were carried out in each instance. In conclusion, an average assessment of the effectiveness of the algorithm may be obtained by merging the outcomes of all k separate iterations.

Results and discussions

Experiments were conducted using a wide variety of medical image types, each presenting its own specific set of challenges, to evaluate the effectiveness of the proposed framework. Specifically, we performed a Leave-one-out cross-validation and compared the effectiveness of our proposed MultiDimensionalU-CNN to that of the conventional U-Net. Throughout the 150 epochs carried out, the better outcomes achieved on the testing dataset were recorded during

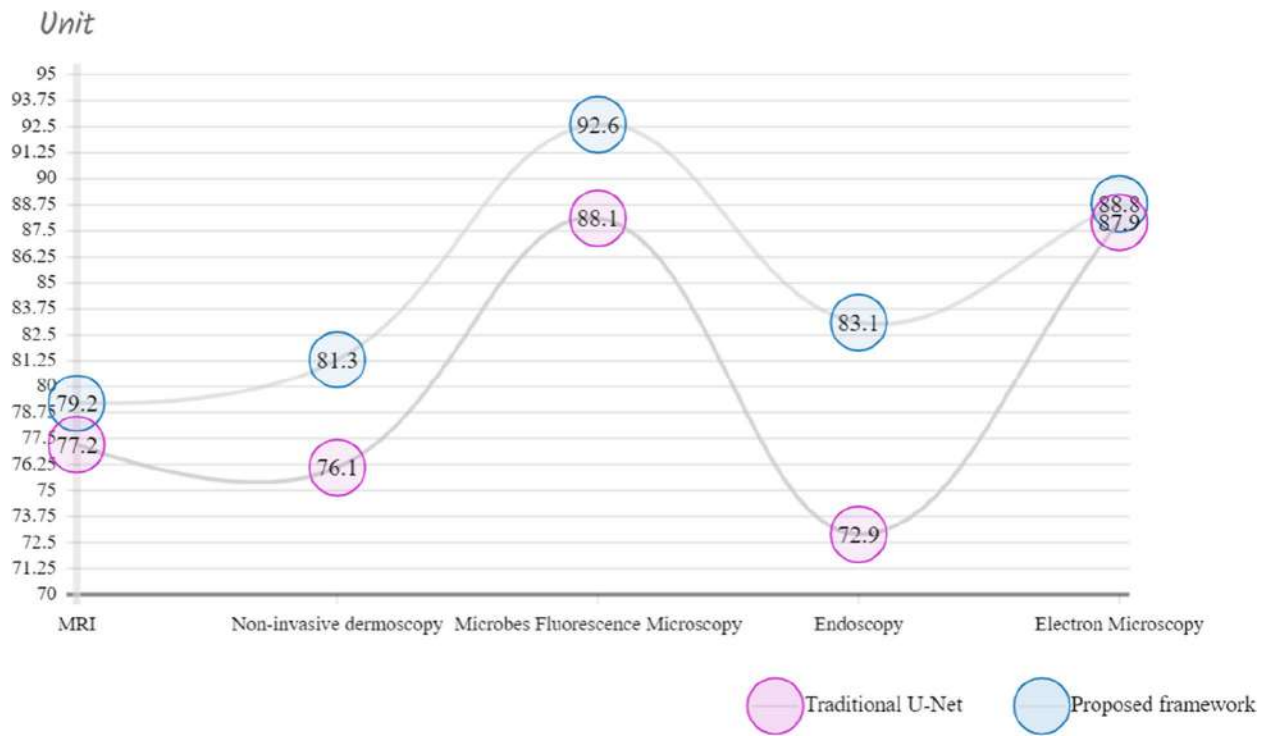


Fig. 7 Graphical chart of comparison outcomes of traditional and proposed MDU-CNN validation accuracy by Leave-one-out cross-validation

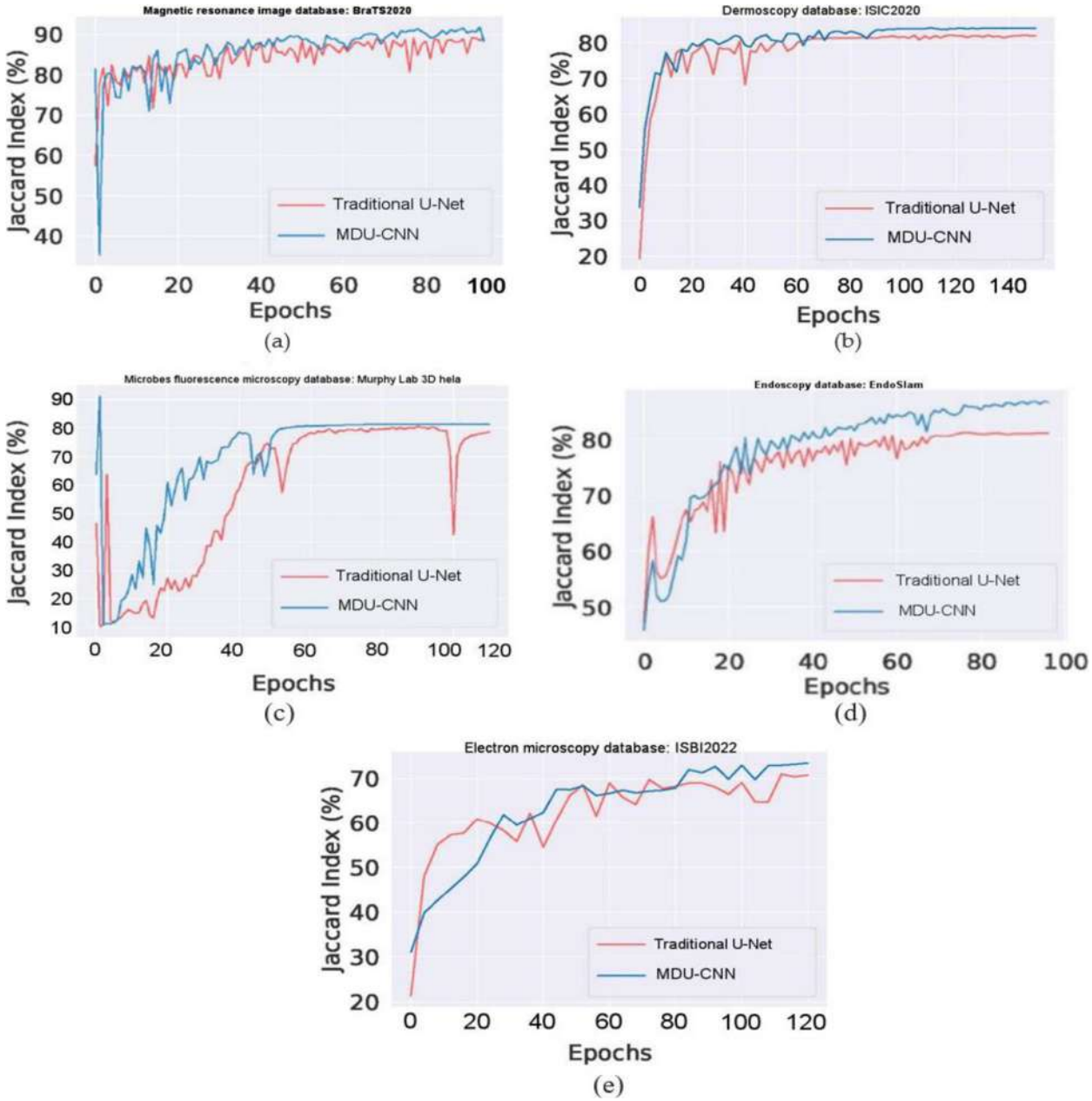


Fig. 8 Improvement in testing efficiency as measured by epoch count **a** MRI; **b** dermoscopy; **c** fluorescence; **d** endoscopy and **e** electron microscopy

each iteration, and these records were aggregated to produce the final result. Table 4 displays the results of the cross-validation using the leave-one-out criterion, applied to each dataset using both the proposed MDU-CNN framework and the classic U-Net model. In this section, we present the most successful results obtained with U-Net and MDU-CNN across all five folds of the datasets. Furthermore, we discuss how MDU-CNN represents a significant improvement over U-Net in terms of quality. It is important to note that the decimal Jaccard indexes have been transformed into percentage values (%) to enhance the readability of the table. Table 4 demonstrates that the performance of our proposed framework surpasses that of the conventional U-Net architecture in the classification of various medical image types. Notably, images obtained through dermoscopy and endoscopy have shown remarkable improvements.

Despite its lower parameter count compared to that of U-net our model achieves great results when it comes to image analysis; specifically, a relative improvement over U-net by 4.5019% for the analysis microscopic fluorescence images and an enhance performance of up to 1·3241% for magnetic resonance imaging (MRI). The depiction of outcomes using Leave-one-out Cross validation for traditional and MDU-CNN can be seen in Fig. 7. To determine which models performed best in every trial we analyzed their overall improvement across all epochs. Figure 8, the outcomes of evaluating all of the datasets based on their effectiveness on the testing dataset at each epoch are demonstrated. Within the context of the cross validation, we have shown the range of Jaccard Indexes at a specific epoch. All the different scenarios shows that the proposed framework gets closer to the right answer much faster. It could be explained by the complementary relationship that exists between batch normalization and residual connections. In addition to this, with the exception of Fig. 8e, the MDU-CNN model has actually outperformed the traditional U-Net model in all of the other instances. Figure 8e shows that the MDU-CNN model, which initially lags behind the traditional U-Net for the electron microscope images, then ultimately converges at a higher accuracy over U-Net. A further interesting result from the studies is that, with the exception of a few insignificant operations, the confidence interval for the efficiency of the MDU-CNN is significantly lower. This finding demonstrates the dependability and consistency of the developed framework. This suggests that the proposed MDU-CNN framework can get better results with less training time than the traditional U-Net design.

U-Net, which is the model that represents the state of the art in terms of medical image segmentation at the moment, has shown quite excellent outcomes in our studies. In Fig. 9, for instance, the U-Net model successfully segments a polyp with a significant Jaccard index due to its well-defined boundary; however,

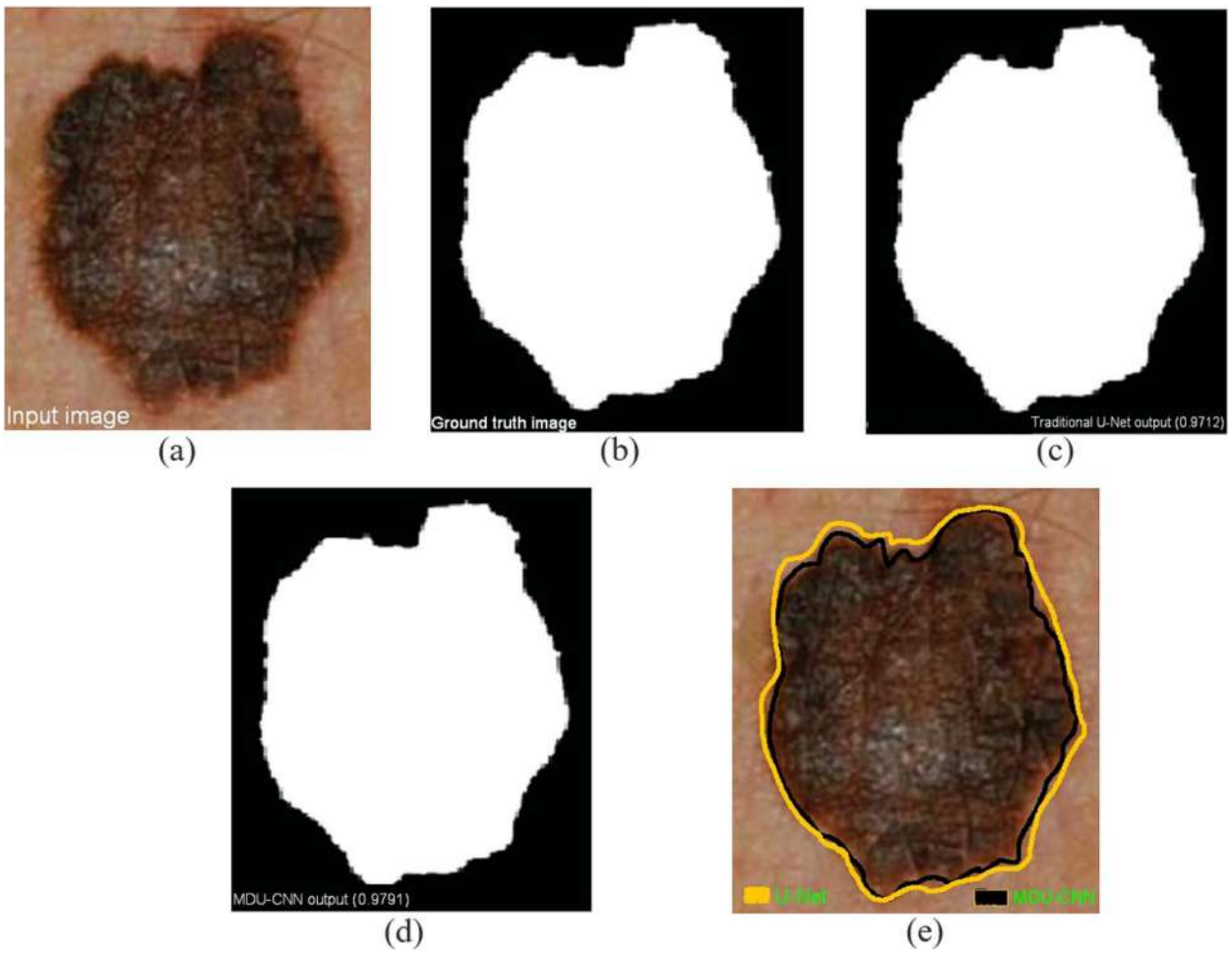


Fig. 9 Polyp with distinct edge identified and segmented **a** input image; **b** ground truth; **c** traditional U-Net; **d** MDU-CNN and **e** final image

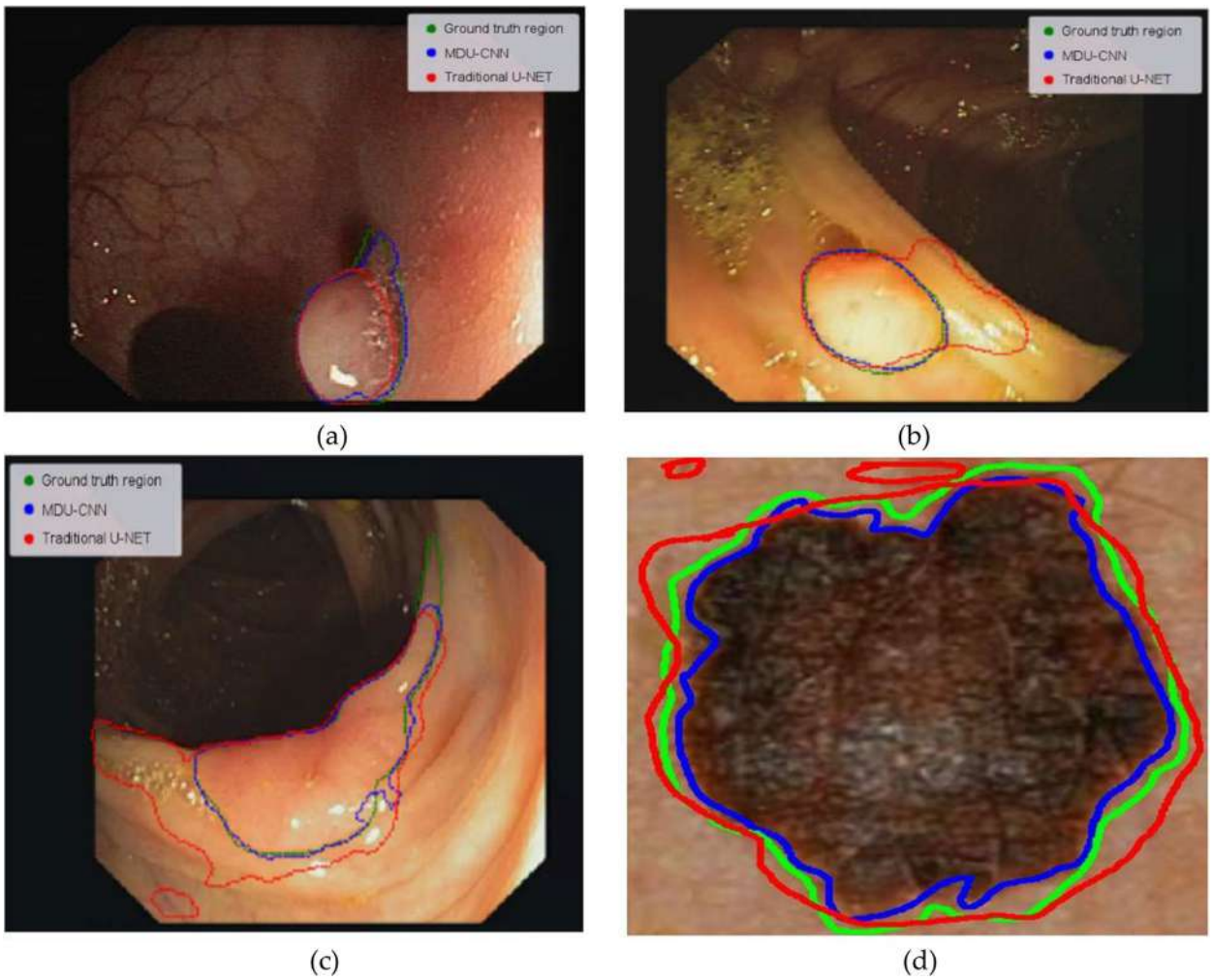


Fig. 10 Boundary-less image segmentation

Table 5 The individual contributions of MultiDimensional blocks and encoder-decoder routes are being studied using ablation. Using 5-fold cross-validation, the EndoSLAM findings are obtained

Type	F = 1	F = 2	F = 3	F = 4	F = 5	Avg
Traditional U-Net	74.11	72.67	73.89	75.22	76.07	74.392
Only encoder-decoder path	76.34	73.78	78.43	75.13	78.21	76.378
Only encoder-decoder convolution block	83.76	77.91	83.32	81.11	82.36	81.692
MDU-CNN	82.25	79.43	83.16	81.02	85.08	82.188

our proposed model performs marginally better. U-Net achieves impressive performance in polyp segmentation (Jaccard index = 0.9712). Relatively improved segmentation can be achieved with MDU-CNN (Jaccard index = 0.9791). U-Net appears to be having some trouble as we analyse more difficult images, especially ones with less apparent boundaries (Fig. 10). Lack of sharp resolution is a common problem in images of colon polyps. U-Net model under-segmented (Fig. 10a) or over-segmented (Fig. 10b) polyps in such circumstances. But in the other side, the proposed MDU-CNN that we proposed performed significantly more effective in both of the instances. Figure 10c displays the results of a comparison between the two models,

showing that MDU-CNN performs better in the circumstances where both models encounter difficulties. Even though dermoscopy pictures often have more clearly defined areas, Fig. 10d shows that MDU-CNN is better at outlining borders. Similarly, it was seen with various different image types. We speculate that MDU-CNN can achieve more pixel-perfect segmentation due to its ability to use a variety of filter sizes. Table 5, illustrates the list of technical short form and its respective abbreviations used in this article.

Ablation study

Ablation research was carried out to evaluate the individual contributions of the MultiDimensional blocks and the encoder-decoder convolution routes. The tests were carried out from two perspectives: in the first, the encoder-decoder convolution paths were simply inserted in a simple U-net based method for our first implementation and replaced one of its two convolutional blocks by using multi-dimensional ones for our second. With the aim of finding the best performing architecture, a comparison among multiple U-net models including those with encoder-decoder convolution routes, multi-dimensional blocks, and MDU-CNN was carried out. The rationale behind selecting the CVC - EndoSLAM dataset for this ablation study is that it proved to be more challenging than any other datasets we have used before. Additionally, findings from a test involving five folds for cross-validation will be shown on table number five. The addition of encoder-decoder convolution routes clearly outperforms the traditional U-Net, as seen in the table. The table clearly shows that the efficiency gains from using multidimensional blocks are considerably greater when they are used independently of an encoder-decoder link. The suggested MDU-CNN, which makes use of both encoder-decoder routes and multidimensional blocks, achieves its best results due to the complementary nature of these two features. Figure 11, depicts the different k-fold cross validation outcome of different models for EndoSLAM dataset.

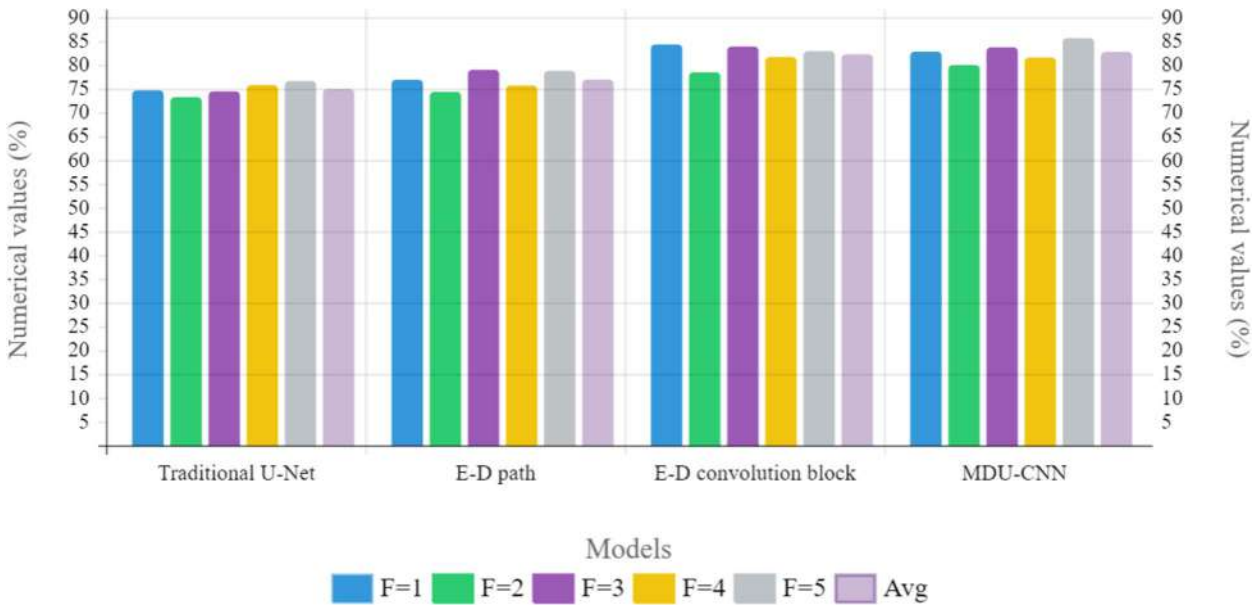


Fig. 11 Graphical illustration of k-fold cross validation of different models from EndoSLAM

Table 6 Data augmenting both models. 5-fold cross-validation yields EndoSLAM results

Type	F = 1	F = 2	F = 3	F = 4	F = 5	Avg
Without DA						
Traditional U-Net	74.11	72.67	73.89	75.22	76.07	74.392
MDU-CNN	82.25	79.43	83.16	81.02	85.08	82.188
With DA						
Traditional U-Net	81.76	77.91	81.32	78.11	79.36	79.692
MDU-CNN	85.25	84.03	88.12	84.02	85.08	85.3

Comment on Data Augmentation (DA)

It is a common understanding that CNNs perform better when they are given a larger dataset. However, up until this point during the presentation all of these results have been obtained without adding extra data. The objective is to assess the general efficiency of MDU-CNN and traditional U-net by evaluating their abilities under similar conditions. Without more data available for training purposes, we have concluded that both models will prove challenging. An added hindrance that needs to be overcome is our initial analysis focused on certain models which incorporated data augmentation. As a result of its proven complexity, we resorted back to working with the EndoSLAM dataset. Our amount of training data was boosted by up to three times due to random flips and rotations or a combination thereof. In assessing their

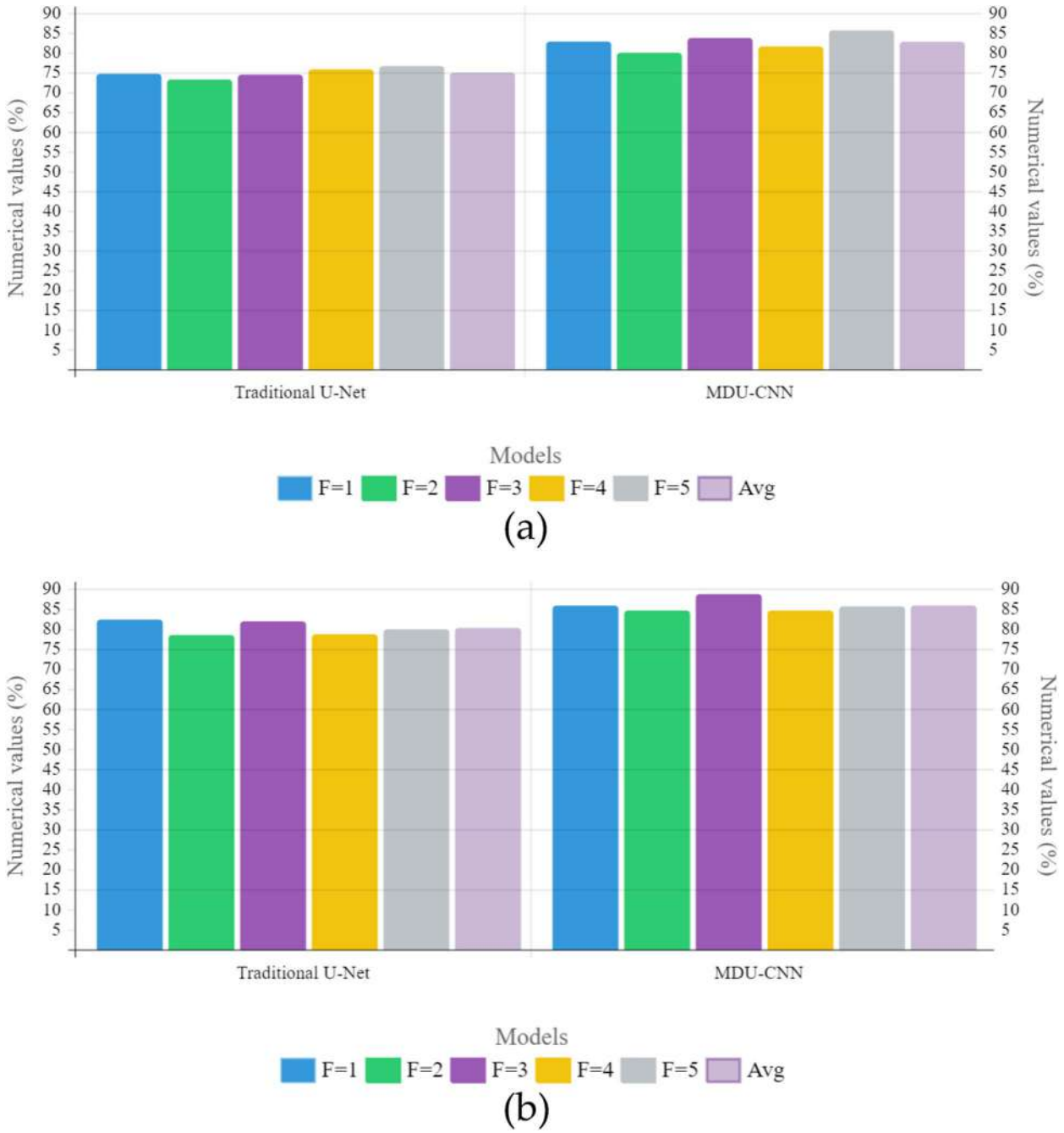


Fig. 12 Graphical illustration of k-fold cross validation of different models from EndoSLAM **a** without data augmentation; **b** with augmentation

performance, we compared how well both models were able to perform. Table 6 provides a comparison of the outcomes obtained with and without DE.

According to the experimental results, both models perform better with data augmentation, and MDU-CNN outperforms standard U-Net. MDU-CNN showed an increase of 3% whereas the improvement of Baseline U-NET model was slightly bigger with 5.31% is the score achieved by the U-net model with a low base line performance scoring only at 74.3%, to fall behind when compared with its counterpart MDU-CNN on this dataset; however increasing data through augmentation has provided some recognition for patterns once learnt easier by counterpart reaching a higher score (82.1%) and in Fig. 12 there is a comparison of the outcomes obtained using different models for Endo-SLAM dataset under different conditions of data augmentation through k-fold cross validation.

Conclusion

In the proposed work, we started by looking closely at the U-Net architecture to see if there were any ways it could be made better. We had assumed that there would be a discrepancy between the features that the encoder model shared and those that the decoder network subsequently propagated. We made encoder-decoder convolution paths, which use some pre-processing techniques to make the mappings of two features more similar to each other. We also proposed multidimensional blocks to add multi-resolution analysis capability to U-Net. We were inspired by the blocks in Inception, and as a result, we were able to design a compact equivalent framework that was lighter and required lesser memory. We proposed a new architecture, the MDU-convolutional neural network, by integrating these improvements. We chose the biomedical image datasets that were extremely distinct from one another and that were made available to the general public. Furthermore, each of those databases has its own distinct difficulty. The cell nuclei in the foreground and the rest of the image in the background are very different in the Murphy Laboratory Fluorescent Microscope database. This could make it the easiest dataset to use for segmentation. In the colon endoscopic images included in the EndoSLAM dataset, distinguishing polyps from the background is typically challenging even for a professional operator. This is a hard dataset to work with because the polyps are so different in shape, size, structure, direction, etc. However, the dermoscopy database that is part of the ISIC2020 challenge includes images that have low contrast to the degree that the skin lesions and the backdrop can often be confused for one another, and vice versa. Both foreground and background textures of varying types, complicates further pattern detection.

ISBI2022's electron microscopy data collection brings a new type of difficulty. As the region being segmented takes up most of the image in this dataset, there is a tendency for the images to be over-segmented. However, the BraTS20 MRI data comprises multimodal 3-dimensional images, which presents a unique challenge. Images should be as close to perfect as possible; segmentation is a challenging task, yet U-Net achieves impressive results. When compared to U-Net, our proposed framework just marginally outperforms it when it comes to these cases. However, the performance improvement by MDU-CNN is significantly greater for complex images that contain things like; noise, perturbations, unclear borders, etc. More precisely, MDU-CNN outperformed U-Net by a factor of 1.32%, 5.19%, 4.50%, 10.23%, and 0.87% across the five datasets shown in Table 4. As a result of these improvements, MDU-CNN segmentations not only score better on the evaluation criteria, but they also look closer to the ground truth. Further, U-Net appeared to oversegment, undersegment, make incorrect predictions, and even overlook the items entirely on the very difficult images. Actually, experimental results showed that MDU-CNN was more consistent and robust. MDU-CNN was capable of recognizing even the smallest subtle of image boundaries; it was durable in segmenting the images with a number of disruptions; and it was removable for outliers. While the U-Net had a tendency to oversegment the majority class, MDU-CNN was able to accurately classify the details. The 3-dimensional version of MDU-CNN also outperformed the 3-dimensional U-Net, which is more than just a 3-dimensional translation of the 2-dimensional U-Net; it provides other improvements and enhancements. To be fair, the proposed MDU-CNN did not always produce perfect segmentations, although it did far better than the standard U-Net. Hence, we agree our proposed MDU-CNN framework can be the possible replacement to the traditional U-Net design.

Future work

In the proposed work, we tried to keep our model's number of parameters about the same as the U-Net model's. In the future, we want to do more tests to identify the best way to combine the method's hyper-parameters. In the near future, we also plan to test our framework on clinical images taken with different methods. In addition to this, we plan to conduct experiments in which our model is subjected to a variety of pre and post processing approaches that are domain and implementation-relevant. As such, we expect that connecting our approach to a domain-specific, professional expertise-based pipeline and combining it with appropriate post-processing steps would further enhance our model's effectiveness and

enable us all to design efficient segmentation approaches for a variety of applications. The successful implementation of Multimodal Biomedical Image Segmentation using a Multi-Dimensional U-Convolutional Neural Network not only advances the field of medical imaging but also paves the way for practical applications with significant implications for diagnosis, treatment, and overall healthcare delivery.

Abbreviations

MDU-CNN	MultiDimensional—Convolutional Neural Network
CAD	Computer-Aided Diagnostic
DL	Deep Learning
ISBI	International Symposium on Biomedical Imaging
U-Net	U-shaped -encode and decode network
LBCDN	Local Binarization Convolutional Neural Network
DC-U-net	Dual Channel Effect U-shaped -encode and decode network
ECA-Net	Efficient Channel Attention
RES-U-NET	Deep Residual UNET U-shaped -encode and decode network
VGG-16UNET	Visual Geometry Group 16
DENSENET	Densely Connected Convolutional Networks
ISIC	International Skin Imaging Collaboration
ReLU	Rectified Linear Unit
2D	2-Dimensional
3D	3-Dimensional
Ph2	Pedro Hispano
HGG	High-Grade Gliomas
LGG	Low-Grade Gliomas
MRI	Magnetic Resonance Image
BraTS2020	Multimodal Brain Tumor Segmentation Challenge 2020
HAM-10000	Human Against Machine with 10000
U2OS	Human Bone Osteosarcoma Epithelial Cells
EndoSLAM	Endoscopic Simulta- Neous Localization and Mapping
ssTEM	Transmission Electron-Microscopy
RAM	Random Access Memory
JI	Jaccard index

Institutional review board statement
"Not applicable".

Informed consent statement
"Not applicable".

Authors' contributions
"Conceptualization, S.S and K.D.; methodology, S.K.M.; validation, D.K and P.K; resources, M.A.S; data curation, P.K; writing—original draft preparation, S.S and K.D ; writing—review and editing, S.K.M, and M.A.S; visualization, S.K.M and P.K; supervision S.K.M and D.K; project administration. S.K.M and M.A.S.

Funding
"This research received no external funding".

Availability of data and materials
The datasets used during the current study are available from the corresponding author on reasonable request.

Declarations

Consent for publication
"Not applicable".

Competing interests
The authors declare no competing interests.

Received: 16 October 2023 Accepted: 9 January 2024
Published online: 08 February 2024

References

1. Ashraf H, Waris A, Ghafoor MF, Gilani SO, Niazi IK. Melanoma segmentation using deep learning with test-time augmentations and conditional random fields. Sci Rep. 2022;12(3948):1–16.
2. Seeja RD, Suresh A. Deep learning based skin lesion segmentation and classification of melanoma using support vector machine (SVM). Asian Pac J Cancer Prev. 2019;20:1555–61.
3. Zhao H, Wang A, Zhang C. Research on melanoma image segmentation by incorporating medical prior knowledge. PeerJ Comput Sci. 2022;8:1–15.
4. Kaur R, GholamHosseini H, Sinha R, Lindén M. Automatic lesion segmentation using atrous convolutional deep neural networks in dermoscopic skin cancer images. BMC Med Imaging. 2022;22(103):1–13.
5. Ahmed N, Tan X, Ma L. A new method proposed to Melanoma-skin cancer lesion detection and segmentation based on hybrid convolutional neural network. Multimedia Tools Appl. 2022. https://doi.org/10.1007/s11042-022-13618-0.
6. Olugbara OO, Taiwo TB, Heukelman D. Segmentation of melanoma skin lesion using perceptual color difference saliency with morphological analysis. Math Problems Eng. 2018;2018:1–19 Article ID:1524286.
7. Nawaz M, Nazir T, Khan MA, Alhaisoni M, Kim JY, Nam Y. MSeg-Net: A Melanoma Mole Segmentation Network Using CornerNet and Fuzzy K-Means Clustering. Comput Math Methods Med. 2022;2022:7502504.
8. Ragab M, Choudhry H, Al-Rabia MW, Binyamin SS, Aldarmahi AA, Mansour RF. 2022. Early and accurate detection of melanoma skin cancer using hybrid level set approach. Front Physiol. 1–15. https://doi.org/10.3389/fphys.2022.965630.
9. Nida N, Irtaza A, Javed A, Yousaf MH, Mahmood MT. Melanoma lesion detection and segmentation using deep region based convolutional neural network and fuzzy C-means clustering. Int J Med Informatics. 2019;124:37–48.
10. Seeja RD, Suresh A. Melanoma segmentation and classification using deep learning. Int J Innov Technol Exploring Eng. 2019;8(12):2667–72.
11. Shifa Kubra N, Divakar HR, Prakash BR. Skin cancer segmentation using U-Net. Int J Res Appl Sci Eng Technol. 2021;9(8):1600–5.
12. Zhao C, Shuai R, Ma Li, Liu W, Menglin Wu. Segmentation of skin lesions image based on U-Net + +. Multimedia Tools and Applications. 2022;81:8691–717.
13. Pennisi A, Domenico B, Suriani V, Nardi D, Facchiano A, Giampetruzzi AR. Skin lesion area segmentation using attention squeeze U-Net for embedded devices. j Digit Imaging. 2022;35:1217–30.
14. Dimšaa N, Paulauskaitė Tarasevičienė A. Melanoma multi class segmentation using different U-Net type architectures. CEUR workshop proceedings: IVUS 2021: Information society and university studies 2021. Proceedings of the 26th international conference on information society and university studies, vol. 2915. 2021. p. 84–91. no.10.
15. Liu L, Mou L, Zhu XX, Mandal M. Skin lesion segmentation based on improved U-Net. IEEE Canadian Conference of Electrical and Computer Engineering. 2019. p. 1–4. https://doi.org/10.1109/CCECE.2019.8861848.
16. Lu H, She Y, Tie J, Xu S. Half-UNet: a simplified U-Net architecture for medical image segmentation. Front Neuroinform. 2022;16:911679 pp.1-10.
17. Salih O, Viriri S. Skin lesion segmentation using local binary convolution-deconvolution architecture. Image Anal Stereol. 2020;39:169–85.
18. Zhen Y, Yi J, Cao F, Li J, Wu J. Skin melanoma segmentation algorithm using dual-channel efficient CNN network. Proceedings of the 5th International Conference on Computer Science and Software Engineering. 2022. p. 549–54. https://doi.org/10.1145/3569966.3570104.
19. Diame ZE, Al-Berry MN, Salem M-M, Roushdy M. Autoencoder performance analysis of skin lesion detection. J Southwest Jiaotong Univ. 2021;56(6):1–10.
20. PrashantBrahmbhatt RC, NathRajan S, BemdMarkscheffel. Skin lesion segmentation using SegNet-U-Net ensemble. Vivechan Internationa/Jaumal of Research. 2019;10(2):22–31.
21. Ma Y, Yang Z. Melanoma recognition and lesion segmentation using multi-instance learning; Springer Nature; 2021. p. 1–17. https://doi.org/10.21203/rs.3.rs-930865/v1.

22. Vimala BB, Srinivasan S, Mathivanan SK, Muthukumaran V, Babu JC, Herencsar N, Vilcekova L. Image noise removal in ultrasound breast images based on hybrid deep learning technique. Sensors. 2023;23:1–16. https://doi.org/10.3390/s23031167.
23. Saravanan S, Kumar VV, Sarveshwaran V, Indirajithu A, Elangovan, Allayear SM. 2022. Computational and mathematical methods in medicine glioma brain tumor detection and classification using convolutional neural Network. Computational and Mathematical Methods in Medicine. Article ID:4380901. pp.1–12. https://doi.org/10.1155/2022/4380901.
24. Saravanan.S, Thirumurugan.P. Performance analysis of glioma brain tumor segmentation using Ridgelet transform and co-active adaptive neuro fuzzy expert system methodology. J Med Imaging Health Inform. 2020;10(11):2642–8.
25. Johri SA, Tripathi A. Parkinson disease detection using deep neural networks. International Conference on Contemporary Computing (IC3). Noida: IEEE; 2019. p. 1–4.
26. Srinivasan S, Dayalane S, Mathivanan SK, Rajadurai H, Jayagopal P, Dalu GT. Detection and classification of adult epilepsy using hybrid deep learning approach. Sci Rep. 2023;13:17574 pp.1-17.
27. Hasib Zunair A, Hamza B. Sharp U-Net: depthwise convolutional network for biomedical image segmentation. Comput Biol Med. 2021;136:1–13.
28. Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby JS. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. Sci Data. 2017;4(170117):1–13.
29. Coelho LP, Shariff A, Murphy RF. Nuclear segmentation in microscope cell images: a hand-segmented dataset and comparison of algorithms. In: Biomedical imaging: From nano to macro. EEE international symposium. IEEE; 2009. p. 518–521.
30. Codella NCF, Gutman D, Celebi ME, Helba B, Marchetti MA, Dusza, Stephen W. Skin lesion analysis toward melanoma detection: a challenge at the 2017 international conference on biomedical imaging, hosted by the international skin imaging collaboration. In: Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International conference. IEEE; 2018. p. 168–172.
31. Bernal J, Sánchez FJ, Fernández-Esparrach G, Gil D, Rodríguez C, Vilariño F. Wm-dova maps for accurate polyp highlighting in colonoscopy: validation vs. saliency maps from physicians. Comput Med Imaging Graph. 2015;43:99–111.
32. Arganda-Carreras I, Turaga SC, Berger DR, Cireşan D, Giusti A, Gambardella LM. Crowdsourcing the creation of image segmentation algorithms for connectomics. Front Neuroanat. 2015;9(142):1–13.

Publisher's Note

CONTRIBUTED PAPER

Deep-Learning-Based 3-D Surface Reconstruction—A Survey

This survey presents a comprehensive overview of these state-of-the-art deep-learning-based approaches to 3-D surface reconstruction.

By Anis Farshian, Markus Götz, *Member IEEE*, Gabriele Cavallaro, *Senior Member IEEE*, Charlotte Debus, *Member IEEE*, Matthias Nießner, Jón Atli Benediktsson, *Fellow IEEE*, and Achim Streit

ABSTRACT | In the last decade, deep learning (DL) has significantly impacted industry and science. Initially largely motivated by computer vision tasks in 2-D imagery, the focus has shifted toward 3-D data analysis. In particular, 3-D surface reconstruction, i.e., reconstructing a 3-D shape from sparse input, is of great interest to a large variety of application fields. DL-based approaches show promising quantitative and qualitative surface reconstruction performance compared to traditional computer vision and geometric algorithms. This survey provides a comprehensive overview of these DL-based methods for 3-D surface reconstruction. To this end, we will first discuss input data modalities, such as volumetric data, point clouds, and RGB, single-view, multiview, and depth images, along with corresponding acquisition technologies and common benchmark datasets. For practical purposes, we also discuss evaluation metrics enabling us to judge the reconstructive performance of different methods. The main part of the document will introduce a methodological taxonomy ranging from point- and mesh-based techniques to volumetric and implicit neural approaches. Recent research trends, both methodological and for applications, are highlighted, pointing toward future developments.

KEYWORDS | 3-D deep learning (DL); 3-D surface reconstruction; geometric DL; geometry processing; machine learning.

I. Introduction

In the last decade, advances in artificial intelligence, in particular in deep learning (DL) [1], [2], [3], have been adopted by a multitude of fields and have, thus, led to major breakthroughs in science and industry alike. One of the major driving forces behind these developments is the field of computer vision, and its desire to "teach" machines how to recognize patterns within image and video data. Initially, a strong emphasis was placed on the interpretation of 2-D information; however, recent advances in cost-effective scanner-based data acquisition and the establishment of large-scale shape repositories have brought the analysis of 3-D data into focus. Still, complexity, variety, and irregularities in 3-D shape representations pose significant methodological challenges.

The reconstruction of 3-D surfaces of objects from different types of input data formats, such as point clouds, depth maps, single-view, or multiview images, is fundamental to a number of application fields, such as computer vision, robotics, CAD, medicine, city planning, disaster prevention, and archeology. One of the special use cases of 3-D reconstruction is human shape reconstructions and pose estimation from images or videos, which is addressed by some other works [4], [5].

Manuscript received 1 March 2022; revised 9 June 2023 and 24 September 2023; accepted 26 September 2023. Date of publication 30 October 2023; date of current version 17 November 2023. *(Corresponding author: Markus Götz.)*

Anis Farshian, **Markus Götz**, **Charlotte Debus**, and **Achim Streit** are with the Steinbuch Centre for Computing, Karlsruhe Institute of Technology, 76344 Karlsruhe, Germany (e-mail: anis.farshian@kit.edu; markus.goetz@kit.edu; charlotte.debus@kit.edu; achim.streit@kit.edu).

Gabriele Cavallaro is with the Jülich Supercomputing Centre, Forschungszentrum Jülich, 52428 Jülich, Germany, and also with the Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, 107 Reykjavik, Iceland (e-mail: g.cavallaro@fz-juelich.de).

Matthias Nießner is with the Visual Computing Laboratory, Department of Informatics, Technical University of Munich, 80333 Munich, Germany (e-mail: niessner@tum.de).

Jón Atli Benediktsson is with the Faculty of Electrical and Computer Engineering, University of Iceland, 102 Reykjavik, Iceland (e-mail: benedikt@hi.is).

Digital Object Identifier 10.1109/JPROC.2023.3321433

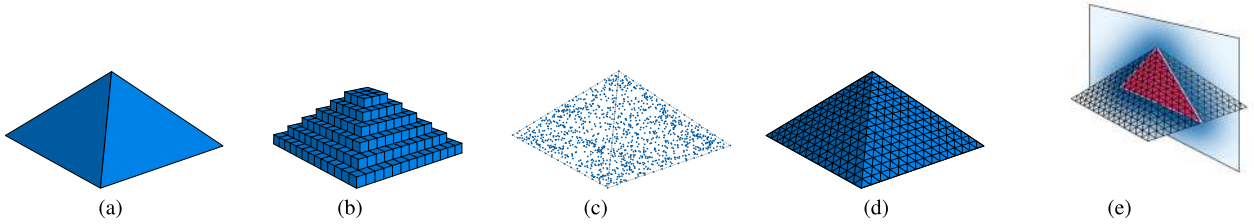


Fig. 1. *Output representations of various 3-D surface reconstruction approaches. DL-based 3-D surface reconstruction approaches can be broadly classified into four main categories according to their representation: volumetric, point cloud, mesh, and an example of implicit neural representation based on SDF. (a) Object. (b) Voxelized. (c) Point cloud. (d) Mesh. (e) Implicit.*

Despite a long research history for 3-D surface reconstruction, the precise representation of 3-D geometrical objects remains an unsolved problem, usually requiring the reconstructed 3-D surfaces to be: 1) highly resolved and smooth; 2) water-tight, i.e., "without gaps"; 3) in accordance with possible ground truth; 4) robust against noisy or incomplete input; and 5) simultaneously, densely, and compressibly represented.

Classical approaches for addressing these problems encompass geometrical or simplistic machine-learning-based algorithms [6], [7]. Most of these methods are not able to comprehensively and consistently reconstruct arbitrary detailed 3-D surfaces. Well-known techniques, such as (screened) Poisson surface reconstruction (PSR) [8], [9], the ball-pivoting algorithm (BPA) [10], and Delaunay triangulation [11], [12], still suffer from scalability issues and struggle to reconstruct fine details for large-scale data.

The recent successes of deep neural networks (DNNs) in other data-driven computational problems, such as classification [13], [14], object detection [14], [15], and segmentation [13], [14], [16], have sparked interest in utilizing DL for 3-D surface reconstruction. Partially overlapping with the latter is the task of shape completion, i.e., enhancing the input data with (partially) occluded shape information.

Several reconstruction-related surveys [17], [18] present early approaches, with [17] providing an overview of the classical and non-DL-based surface reconstruction methods from point clouds with respect to priors and [18] reviewing RGB-D scene reconstruction approaches. There is another DL-based surface reconstruction survey [19] with a focus on image-based methods. This article, however, covers broader data modalities and reviews recent trends in 3-D surface reconstruction including implicit neural representation and neural radiance fields (NeRFs) thoroughly.

Therefore, the fast-paced development of the field makes it, however, necessary to revisit up-to-date research frequently. The current landscape of DL-based 3-D reconstruction can be broadly classified into four main categories according to their representation, as depicted in Fig. 1: 1) volumetric, i.e., representing a surface with small cuboids, either a dense 3-D voxel grid [20], [21], [22], [23], [24], [25] or an octree [26], [27], [28], [29]; 2) point-based [30], [31], [32], [33], i.e., using points to present a surface; 3) mesh-based [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], i.e., describing an object with vertices, edges, and faces; and 4) implicit neural representation [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], i.e., representing a shape as a neural network that takes any (x, y, z) coordinate as input and maps it to an occupancy or signed distance of the shape at that coordinate or modeling radiance or appearance properties of an object such as NeRF-based approaches [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68].

In this survey, we present a comprehensive overview of these state-of-the-art DL-based approaches to 3-D surface reconstruction. Our main goal is to provide method researchers with a guide to current work and applied researchers with a toolbox for their domain challenges. Toward this end, we first provide a broad introduction to input data formats (see Section II), acquisition technologies (see Section III), and widely used benchmarking datasets (see Section IV). Section V covers evaluation metrics enabling to quantitatively judge the reconstructive performance of a method, independent of being classical or learning-based. The main part of this survey (see Section VI) highlights DL methods to reconstruct 3-D surfaces using volumetric, point- and mesh-based, and implicit neural representations. We assume that the reader has a general grasp of neural networks and DL concepts to thoroughly follow the content. Discussion, current trends, and challenges are highlighted in Section VII. Finally, Section VIII summarizes and concludes this survey.

II. INPUT DATA

Various types of data representations can be used as input for the 3-D surface reconstruction task. Conventional representations of 3-D inputs can be divided into Euclidean and non-Euclidean data. Examples of non-Euclidean data representations are point clouds or meshes, while Euclidean data representations can be volumetric, RGB-D data, or multiview images.

Point clouds are currently the most common format of raw 3-D sensor data. With the improvement of scanning devices, leading to enhanced capabilities for capturing the

surrounding 3-D environment in various applications and representing it with points, point clouds are becoming increasingly important and available. Thus, processing this type of representation using neural networks and DL techniques has attracted considerable attention. From a mathematical point of view, point clouds comprise an irregular data structure in the form of an unordered set of points. Each point on a 3-D surface of an object can basically be defined by a vector of its (x, y, z) coordinates, which can be inferred by various 3-D data acquisition techniques. Hence, the size of the representation matrix of a 3-D object is initially $N \times 3$ for N points. The matrix may also contain different properties including color, transparency, surface normals, and other scanner information. However, pure point clouds do not include the interconnections between vertices. Since a point cloud is a set, its elements are orderless, a characteristic that causes many challenges for surface reconstruction methods. Point clouds can be easily converted to/extracted from other data representations, such as voxels, depth maps, or meshes, and vice versa. Furthermore, they can be extracted from depth images by projecting the depth value of each pixel into 3-D space.

Meshes are another highly popular type of representation for 3-D objects providing detailed and connected geometries in an efficient way. They are irregular data embedding in continuous space. Their basic components are vertices, edges, i.e., pairs of vertices, and (triangular) faces, i.e., n-tuples of edges, forming an undirected graph.

In volumetric representations, the basic element is a voxel. A voxel in a 3-D grid is a cuboid equivalent to a pixel in 2-D space. The 3-D grid, regardless of being sparse or dense, can be fed to a neural network as the input.

An RGB-D image is a combination of an RGB image and a depth image. It not only has RGB information for each pixel but also includes depth information.

Multiview images are a collection of (single-view) images taken from different angles of an object. By putting these images together, 3-D information can be partly retrieved.

On the other hand, 2-D data, such as single-view RGB images, can also be considered the input to a network for surface reconstruction individually, in which the method is called single-view reconstruction (SVR) [69], [70], [71] or in conjunction with another 3-D input mentioned earlier.

III. DATA ACQUISITION

As explained in Section II, point clouds are the most common format of raw 3-D sensor data. 3-D point cloud data are acquired through sensing technologies that measure distance [i.e., 3-D laser scanning also known as light detection and ranging (LiDAR)] or generated with stereo- and multiview image-derived systems that can be based on red, green, blue-depth (RGB-D) cameras, stereo cameras, and multiple synthetic aperture radar (SAR) image pairs [16], [72]. High-quality 3-D point clouds can capture the 3-D surface geometries of target objects (e.g., physical features that occupy the Earth's surface and ocean bottom) with a spatial accuracy up to the millimeter level and a point density of a few thousand points per square meter (pts/m^2).

A. 3-D Laser Scanning (LiDAR)

LiDAR is a remote sensing (RS) active technology that uses light in the form of a pulsed laser to measure the distance between the sensor and the object under study [73]. By measuring the time that emitted pulses take to travel to a target, LiDAR derives 3-D representations of objects. LiDAR can also operate at different wavelengths (i.e., multispectral LiDAR [74], [75]) to discriminate the different spectral reflectance of land-cover classes [76], [77].

Depending on the platform on which the LiDAR sensor is mounted, a 3-D laser scanner is classified as a terrestrial laser scanner (TLS or ground LiDAR), airborne laser scanner (ALS), mobile laser scanner (MLS), and unmanned laser scanner (ULS) [16], [72].

A TLS uses ground-based RS systems (e.g., tripods) to cover middle- or close-range areas with scans performed in all directions, including upward [78]. Once scans of a single zone are completed, the tripod is moved to another location to scan from another angle or capture data from a new area. As TLS systems are static during the acquisition process, they reach the highest point cloud density and can produce high-quality 3-D models of the interiors of buildings and heritage sites.

Nevertheless, TLS systems cannot always be used, especially for scanning restricted locations that are not safe or accessible for teams (e.g., areas of dense vegetation and unsafe building sites). In these cases, LiDAR sensors can be mounted on airborne platforms. ALS systems are also used to acquire point cloud data over large areas (e.g., for 3-D building reconstruction [79]).

When target regions are directly accessible, their structures and objects can be reconstructed from data acquired by MLS systems, i.e., LiDAR sensors mounted on moving vehicles (e.g., to derive high-resolution 3-D city models [80]).

Since drones and other unmanned vehicles have become cheaper and autonomous navigation more reliable [81], ALS and MLS are often operated as ULS systems. Their platforms are compact and lightweight, which enables them to be exploited as first responders for disaster management. ULS systems can make a first scan of the terrain to track movements and changes, and deliver 3-D mapping of the most affected locations [82], [83].

B. Photogrammetry

While LiDAR performs a direct measurement of the target object, i.e., by physically hitting a feature with light and measuring the reflection, approaches based on photogrammetry or computer vision theory [84] use a set of overlapping images taken from different locations to identify isolated points within a target. This includes not only airborne photogrammetry but also satellite stereo

systems, which can map larger regions quickly. Image-based reconstruction algorithms can estimate the relative locations of these points and eventually convert the overlapping images into a 3-D point cloud. For instance, the structure from motion (SfM) algorithms [85] can process multiview images simultaneously through estimating camera positions and orientations automatically, while dense matching and multiview stereo (MVS) algorithms [86] can generate a large volume of point clouds (e.g., large-scale scenarios and crowded environments).

C. RGB-D Camera

Similar to LiDAR, RGB-D cameras measure the distance between the sensor and the objects. Depth information of each RGB pixel of the image is retrieved via a depth sensor. An RGB-D camera generates a colored point cloud by mapping RGB images with depth information (i.e., images include the (x, y, z) spatial coordinates and RGB colors). In this case, the point cloud is not the direct result of RGB-D scanning [87], [88] since the camera generates pixelwise depth data rather than unstructured points. RGB-D cameras are generally cheaper than LiDAR systems and are mostly used in indoor environments for close-range applications [89].

Structured light and Time of Flight (ToF) [90], which are active imaging systems, serve as depth cameras and calculate the distance from the sensor to an object, consequently providing 3-D information. The depth of an object can be determined using ToF sensors by measuring the duration of light travel from the sensor to the object and back. By determining the ToF of light, these sensors can calculate the object's distance and create a detailed depth map, which can be directly used or easily converted to a point cloud for instance. Structured light sensors employ the deformation of a projected pattern to determine the distance. By emitting a known light pattern onto a scene and examining how the pattern changes as it interacts with objects in the scene, these sensors are able to accurately measure the depth information of the objects. Structured light technology-based 3-D scanners are comparatively more affordable, being lighter in weight than their laser-based counterparts as well. Due to their higher degree of sensitivity to lighting conditions, they may not operate well in outdoor environments or in challenging conditions such as dusty rooms. For black or glossy surfaces, a specific spray should be applied before 3-D scanning.

D. SAR Point Cloud

SAR is an active RS system that can operate day and night and can penetrate clouds and smoke. Interferometric SAR (InSAR) extends the principle of SAR to the 3-D domain [91] by taking advantage of the physical properties of microwaves [92]. An InSAR system compares the phase of multiple SAR image pairs acquired from slightly different viewing angles to generate InSAR-based point clouds. The SAR tomography (TomoSAR) and persistent scatterer interferometry (PSI) are two major techniques that generate point clouds with InSAR [16]. They are used to monitor terrain changes (e.g., surface deformations and human-made structures [93]).

E. Videogrammetry

3-D point clouds can also be reconstructed using video frames (i.e., the input data are video streams instead of a collection of images). This approach is referred to as videogrammetry [94] and is based on the principles of photogrammetry. It can reconstruct point clouds from the frames of a video since their information is sequentially interconnected. Videogrammetry approaches provide a valuable alternative to camera images. They can be semi-automatic since the search for target points in different images can be achieved by measuring or tracking features of interest between consecutive video frames. However, the reconstruction needs to be coupled with effective frame selection algorithms (e.g., video frames are selected based on the surveyed geometry) and robust 3-D processing methodologies [95].

IV. DATASETS

DL approaches are data-demanding; thus, they require large amounts of data with high-quality 3-D shapes and ground truths. Recent developments in scanning and sensing technologies have led to the collection of various widely used and openly accessible benchmarking datasets. These datasets are used to train and evaluate the performance of DL methods for different tasks, including 3-D reconstruction. In this section, we summarize some of the most popular datasets, which can be used by different 3-D DL approaches, with a focus on 3-D reconstruction. Table 1 offers a comparative overview of these datasets.

1) ShapeNet [96] is a richly annotated, large-scale synthetic dataset of 3-D shapes represented by 3-D computer-aided design (CAD) models of objects, providing roughly 3 000 000 shapes. This dataset has been used for computer graphics and vision purposes. The full ShapeNet dataset is not yet publicly available. It consists of several subsets, including ShapeNetCore and ShapeNetSem. ShapeNetCore contains a single clean 3-D shape that covers 55 common object categories with about 51 300 unique 3-D shapes. ShapeNetSem is a smaller, more densely annotated subset, containing 12 000 shapes of a broader set of 270 categories. For each shape in ShapeNet, annotations such as its geometry, texture, parts, symmetry planes, voxelization, screenshot, category, alignment, and size are available. The final representation of an object in this dataset can be a 3-D mesh. The 3-D shapes are stored in the Wavefront object file format (`.obj`), which describes the surface geometry of a 3-D shape and includes vertices and faces, along with material template library (`.mtl`) files used to store material definitions. An `.mtl` file is a companion file

Table 1 Comparison of Benchmark Datasets

Name	Count/Size	Dataset Type	Representation	Scene Type	Source	DL Tasks
ShapeNetCore	51,300 3D models from 55 object categories	Synthetic	Mesh	Indoor and outdoor objects	CAD model	Shape recognition, reconstruction, retrieval
ShapeNetSem	12,000 3D models of 270 object categories	Synthetic	Mesh	Indoor and outdoor objects	CAD model	Shape recognition, reconstruction, retrieval
PartNet	573,585 part instances of 26,671 3D ShapeNet models in 24 object categories	Synthetic	Mesh and point cloud	Indoor object parts	CAD model	Part-level understanding
ModelNet	127,915 3D models with 662 object categories	Synthetic	Mesh	Indoor and outdoor objects	CAD model	Recognition, reconstruction, generation, and completion
KITTI	Around 49,000 frames from 5 categories	Real-world	Image and point cloud	Outdoor	RGB and LiDAR	Stereo, optical flow, visual odometry, SLAM, 3D object detection, and object tracking
Semantic KITTI	23,201/20,351 scans with 4549 points from 28 classes	Real-world	Point cloud	Outdoor	LiDAR (MLS)	Semantic segmentation and scene completion
ScanNet	2,5 million frames from 1500 RGB-D scans	Real-world	Image and mesh	Indoor	RGB-D Sensor	Object classification, voxel labeling, model retrieval, and reconstruction
Matterport3D	10,800 views from 90 scenes	Real-world	Image and mesh	Indoor	RGB-D Sensor	Scene understanding, normal prediction, classification, semantic segmentation, and reconstruction
NYU depth v2	1449 RGB-D images consisting 464 diverse scenes across 26 scene classes	Real-world	Image	Indoor	RGB-D	Segmentation
Sun3D	415 sequences captured for 254 different spaces in 41 different buildings	Real-world	Image and point cloud	Indoor	RGB-D sensor	Scene understanding, reconstruction, and segmentation
Sun RGB-D	10,335 RGB-D images from 47 scene categories consisting about 800 object categories	Real-world	Image	Indoor	RGB-D sensor	Scene understanding, semantic segmentation, object detection, orientation, and classification
Sydney urban objects	631 scans from 26 object categories	Real-world	Point cloud	Outdoor	LiDAR	Classification and recognition
ABC	1 million 3D models	Synthetic	Mesh	Indoor	CAD model	Feature detection, shape reconstruction and surface normal estimation
Semantic3D	4 billion points in 8 class labels	Real-world	Point cloud	Outdoor	LiDAR (TLS)	Classification and semantic segmentation
H3D	Around 73 million points and 3,5 million faces in 11 classes	Real-world	Point cloud and mesh	Outdoor	LiDAR	Semantic segmentation

Table 1 *(Continued.)* Comparison of Benchmark Datasets

3D-Front	18,968 rooms with 13,151 furniture objects from 31 scene categories	Synthetic	Mesh	Indoor	CAD model	3D scene understanding, reconstruction, and segmentation
3D-FUTURE	20,240 images of 5,000 different rooms	Synthetic	Mesh	Indoor	CAD model	2D instance segmentation, 3D object pose estimation, image-based 3D shape retrieval, 3D reconstruction from a single image, and texture recovery for 3D shape
SensatUrban	4 billion points in 13 semantic class labels	Real-world	Image and point cloud	Outdoor	UAV Photogrammetry	Urban-scale point cloud understanding

for one or more `.obj` files, which describes some surface appearance properties.

2) PartNet [97] is a dataset of 3-D objects, built on top of ShapeNet with fine-grained, hierarchical, and instance-level 3-D part annotations. The dataset comprises 573 585 part instances of 26 671 ShapeNet 3-D shapes in 24 indoor object categories in an attempt to enable part-level understanding of 3-D objects.
3) ModelNet [98] is a large-scale CAD model synthetic dataset. It includes a comprehensive and clean collection of 127 915 CAD models with 662 object categories and consists of two subsets, ModelNet10 and ModelNet40 with ten and 40 classes, respectively. ModelNet10 has also been annotated with the orientation of the CAD models, which are given in the Geomview object file format (`.off`). The final representation of this dataset can be a mesh.
4) KITTI [99], [100] is a real-world urban scene dataset composed of images and point clouds. The dataset was acquired by the autonomous driving platform Annieway while driving around the city of Karlsruhe. Evaluation benchmarks were developed for several computer vision and robotic tasks, such as stereo, optical flow, visual odometry, SLAM, 3-D object detection, and 3-D object tracking. Semantic KITTI [101], which is based on KITTI, provides pointwise annotations for semantic segmentation and semantic scene completion purposes. The dataset comprises 28 classes including classes for nonmoving and moving objects.
5) ScanNet [102] is a 3-D reconstruction dataset of indoor scenes consisting of 2.5 million frames (views) derived from more than 1500 RGB-D scans. 3-D camera poses, surface reconstructions, and instance-level semantic segmentations are also provided. All scans are reconstructed into 3-D mesh models. The data are stored in polygon file format (`.ply`).
6) Matterport3D [103] is another dataset facilitating RGB-D scene understanding. It captures 10 800 panoramic views from 194 400 RGB-D images of 90 building-scale scenes. The dataset is annotated with surface reconstructions as textured meshes, camera poses, and 2-D/3-D semantic segmentations.
7) NYU depth v2 [104] introduced an annotated dataset of 1449 RGB and depth images, consisting of 464 diverse indoor scenes. These images were acquired by RGB and depth cameras from Microsoft Kinect.
8) Sun3D [105] is a real-world large-scale dataset of RGB-D frames with semantic object segmentations and camera pose used for scene understanding. It consists of 415 sequences captured for 254 different indoor spaces in 41 different buildings.
9) SUN RGB-D [106] is a dataset containing over 10 000 RGB-D images from NYU depth v2 [104], Berkeley B3DO [107], and SUN3D [105] datasets. These images are annotated with 2-D segmentations (146 617 2-D polygons), 3-D object boxes (64 595 3-D bounding boxes), 3-D room layout, 3-D object orientation, and scene category for each image.
10) The Sydney Urban Objects dataset [108] is a point cloud dataset that contains 631 scans of 26 different object classes, including vehicles, pedestrians, trees, and signs taken in the city of Sydney. Each object's information is available in three file formats: ASCII CSV format (`.csv`), binary-packed CSV (`.bin`), and meta information files (`.meta`).
11) The ABC dataset [109] is a CAD model dataset with one million 3-D models. Koch et al. [109] offered a pipeline that is able to convert these CAD models into other representations in order to be processable by DL techniques. These models are provided in `.obj` and 3-D systems' stereolithography CAD file format (`.stl`).
12) Semantic3D.net [110] is a large labeled 3-D point cloud dataset of natural scenes with over four billion points in eight class labels. These dense point clouds, which were recorded by TLSs, depict urban and rural outdoor terrestrial scenes.
13) H3D [111] is a high-resolution real-world dataset containing both point clouds (H3D(PC)) and meshes (H3D(Mesh)) of airborne LiDAR data and can be used for semantic segmentation in geospatial applications.

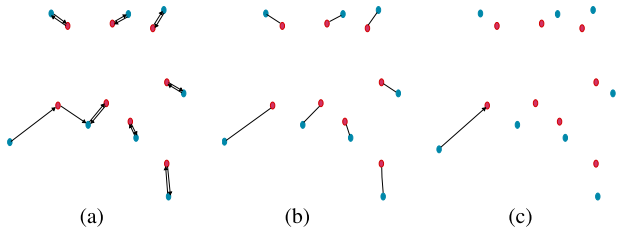


Fig. 2. *Visualization of (a) CD, (b) EMD, and (c) HD metrics. Red dots and blue dots belong to two different point sets, and each of these metrics measures the distance between these two sets in a unique way.*

The point clouds are classified into 11 classes, and labeled 3-D textured meshes can be derived from them.

14) 3-D Furnished Rooms with layOuts and semaNTics (3D-Front) [112] is a synthetic dataset of indoor CAD model scenes, containing 18 968 rooms with 3-D objects. The individual objects are taken from 3D-FUTURE [113]. The CAD models are stored in `.obj` and `.mtl` file formats.
15) 3-D Furniture shape with TextURE (3D-FUTURE) [113] is a repository of 3-D furniture shapes in the household scenario enriched with 3-D and 2-D annotations. It includes 20 240 synthetic images of 5000 different rooms. Stylistic and texture details of individual objects are provided. The 3-D models are stored in `.obj` file format.
16) SensatUrban [114] is a dataset for urban-scale point cloud understanding. It covers 7.6 km^2 of urban areas in Birmingham, Cambridge, and York cities. The point clouds are obtained from high-resolution aerial images, which are captured by the UAV mapping system.
17) Stanford 3-D Scanning Repository [115] is a surface reconstruction repository containing some famous 3-D models, such as the Stanford bunny, happy Buddha, dragon, and armadillo in `.ply` format. These 3-D models and some others also exist in the Large Geometric Models Archive [116].

V. EVALUATION METRICS

Evaluation metrics are used to assess the performance of DL models [1], [2], [3]. Various metrics have been proposed for testing deep geometric learning methods. Some of the common distance metrics used for surface reconstruction methods are Chamfer distance (CD), earth mover's distance (EMD), and Hausdorff distance (HD), which all measure the discrepancy between two sets, as illustrated in Fig. 2. Another common metric for evaluating 3-D reconstruction solutions is the Intersection over Union (IoU). Furthermore, the formulas in this section denote false positives, false negatives, true positives, and true negatives as FPs, FNs, TPs, and TNs, respectively.

1) The CD [30] measures the distance between two different surfaces or sets of points by first calculating the distances between predicted points and their ground-truth nearest neighbors and then averaging all of these distances. The calculated value represents the dissimilarity between predicted output and ground truth. The lower the value, the better the result. Let S_1 and S_2 be two point clouds that represent the predicted and ground-truth shapes, and x and y be two points that belong to these point clouds, respectively. Then, the CD is defined as

$$d_{\mathrm{CD}}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} ||x - y||_2^2 + \sum_{y \in S_2} \min_{x \in S_1} ||x - y||_2^2. \quad (1)$$

2) The EMD, also known as the Wasserstein distance in mathematics and optimization theory) [30], [117], [118], is based on solving an optimization problem, called the transportation problem. The transportation problem attempts to find the least-expensive flow of goods from suppliers to consumers while satisfying the consumers' demand. For the calculation of the EMD of two point sets, each point in one set should be assigned to a unique point in the other set to fulfill optimal assignment. EMD uses bijection between the points that minimize the total sum of the pairwise distances. Consider $S_1 \subseteq R^3$ and $S_2 \subseteq R^3$ to be two point sets of equal size, representing the predicted and ground-truth shapes, respectively. The EMD [30] is defined as

$$d_{\mathrm{EMD}}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} ||x - \phi(x)||_2 \quad (2)$$

where $\phi : S_1 \to S_2$ is a bijection.

3) The HD considers the farthest and largest dissimilarity between predicted output and ground truth. A point in one set that has the worst mismatch and maximum distance from its nearest point in the other set determines the HD

$$d_{\mathrm{HD}}(S_1, S_2) = \max \left\{ \max_{x_i \in S_1} \min_{y_j \in S_2} ||x_i - y_j|| \; \max_{y_j \in S_2} \min_{x_i \in S_1} ||x_i - y_j|| \right\}. \quad (3)$$

The metric is, however, not very robust toward outliers.

4) The IoU, also known as the Jaccard Index, is often used as a quality measure in object detection and semantic segmentation. As illustrated in Fig. 3, it is defined as the overlap between the prediction and the ground truth, divided by their union. The lower the IoU, the worse the prediction result.

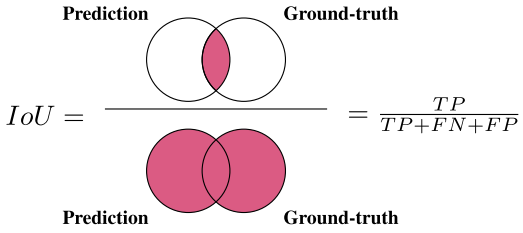


Fig. 3. *Visual intuition of the IoU metric.*

IoU can also be easily utilized for evaluating voxel-based representations and specifying the overlap between a reconstructed 3-D voxel and its voxelized ground truth. For volumetric approaches, IoU can be formulated as [20]

$$\text{IoU} = \frac{\sum_{i,j,k}\left[I\left(p_{(i,j,k)} > t\right) I\left(y_{(i,j,k)}\right)\right]}{\sum_{i,j,k}\left[I\left(I\left(p_{(i,j,k)} > t\right) + I\left(y_{(i,j,k)}\right)\right)\right]} \quad (4)$$

where $I(.)$ is an indicator function, $p(i,j,k)$ is the predicted voxel occupancy probability, t is a voxelization threshold, and $y(i,j,k)$ is the ground-truth occupancy probability.

5) In classification problems, precision is the number of predictions correctly assigned to one label, i.e., true positives, divided by the number of all predictions assigned to that label, including those identified incorrectly, i.e., false positives (see Fig. 4)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (5)$$

The average precision (AP) is computed by averaging all precision values of all positively labeled samples [98]. The mean AP (mAP) is the average of AP calculated over all classes. For point clouds, precision is calculated as the percentage of predicted points that are close to the ground-truth surface, i.e., with a distance less than a specific threshold [119].

6) Recall or sensitivity denotes the ratio between the number of predictions correctly assigned to one class (TP) and the actual number of elements in that class, including those that are incorrectly assigned to the other label (FN) (see Fig. 4). It is a measure of how well a DL model can find all labels of one class

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (6)$$

For point clouds, recall is calculated as the percentage of points on the ground truth, which are close to the predicted surface, i.e., having a distance less than a specific threshold [119].

7) The F_1 score, also known as balanced F-score, F-measure, or dice similarity coefficient (DSC), is the harmonic mean of precision and recall. The higher the value, the better the result

$$F_1\text{score} = 2 * \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}. \quad (7)$$

For point clouds, precision and recall can be calculated by checking the percentage of points in one point cloud, for instance, the predicted point cloud or the ground truth, which can find a neighbor from the other point cloud within a threshold [38]. Intuitively, the F-score can be interpreted as the percentage of points that were reconstructed correctly [119].

8) In classification problems, the accuracy (Acc) is the ratio between correct predictions and all predictions, i.e., it shows how much of the data is labeled correctly

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{FN} + \text{TN})}. \quad (8)$$

However, it is not an appropriate metric for imbalanced datasets as it does not take into account the distribution skew [120].

9) Normal consistency (NC) [45] is defined as the mean absolute dot product of the surface normal of each point, i.e., a perpendicular vector to the surface at the given point, in one mesh, and the surface normals of its nearest neighbors in the other mesh

$$\begin{aligned}\text{NC}\left(\hat{M}, M\right) &= \text{Normal Consistency}\left(\hat{M}, M\right) \\ &= \frac{1}{2|\partial\hat{M}|}\int_{\partial\hat{M}} |\langle n(p), n(\pi_2(p))\rangle| dp \\ &+ \frac{1}{2|\partial M|}\int_{\partial M} |\langle n(\pi_1(q)), n(q)\rangle| dq\end{aligned} \quad (9)$$

where $\partial\hat{M}$ and ∂M are predicted and ground-truth mesh surfaces, $n(p)$ and $n(q)$ are unit normal vectors on these mesh surfaces, respectively, $\pi_2(p)$ and $\pi_1(q)$ indicate the projections of p and q on the aforementioned surface meshes, respectively, and $\langle .,. \rangle$ implies the inner product. The higher the NC, the better the result.

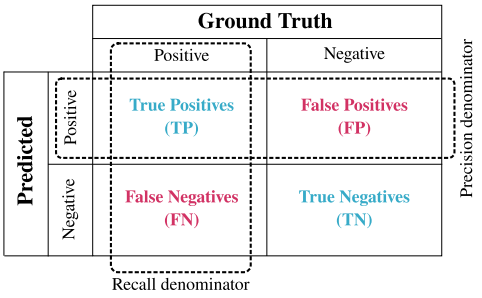


Fig. 4. *Confusion matrix for binary classification.*

10) The Jensen–Shannon divergence (JSD) [31] measures the similarity between marginal point distributions. It is mainly based on the Kullback–Leibler (KL) divergence [121]. Considering two point clouds and a voxel grid that discretizes 3-D space, the number of points within each voxel from the predicted point set P and the ground-truth point set G is counted. The JSD between the obtained empirical distributions (P_P, P_G) is calculated as

$$\text{JSD}\left(P_P||P_G\right) = \frac{1}{2}D_{\text{KL}}\left(P_P||M\right) + \frac{1}{2}D_{\text{KL}}\left(P_G||M\right) \tag{10}$$

where $M = (1/2)(P_P + P_G)$.

11) Coverage [31] quantifies the fraction of points in the ground-truth set S_2, which are matched to points in the predicted set S_1. A match happens when the nearest neighbor in the ground-truth set is found for each point in the predicted set

$$\text{Coverage}\left(S_1, S_2\right) = \arg\min_{Y \in S_2} \frac{D\left(X, Y\right)|X \in S_1|}{|S_2|} \tag{11}$$

where $D(.,.)$ or "nearness" is measured using distance metrics, such as CD or EMD. High coverage indicates that most of the points in S_2 are roughly present within S_1. However, this does not assess the quality of the predicted set. Achieving perfect coverage is possible, despite large distances between the predicted point set and the ground-truth set [33].

12) Minimum matching distance (MMD) [31], [33] is a complement to the coverage metric. It measures the distance between every point in the ground-truth set S_2 and its nearest neighbor in the predicted set S_1 and averages these distances in order to evaluate the quality of the predicted set

$$\text{MMD}\left(S_1, S_2\right) = \frac{1}{|S_2|}\sum_{Y \in S_2}\min_{X \in S_1} D\left(X, Y\right) \tag{12}$$

where $D(.,.)$ is measured using distance metrics such as CD or EMD.

13) Light field descriptor (LFD) [122] measures visual similarity between 3-D shapes. In short, LFD assumes that a 3-D object can be represented as a number of 2-D views; therefore, if two 3-D models are similar, they also look alike from all views. A light field, which is used in image-based rendering, is defined as a 5-D function that represents the radiance at a given 3-D point along a given direction. To extract LFD for a 3-D model, a set of image renderings (silhouettes) are obtained from different angles. These rendered images are acquired using cameras located on the vertices of a fixed regular dodecahedron, i.e., 20 vertices, which surrounds the 3-D model. Each of these silhouettes is then encoded both by a region shape descriptor (Zernike moments descriptor) and a contour shape descriptor (Fourier descriptor) for similarity comparisons. A visual representation can be found in Fig. 5. LFD is a good visual similarity metric for 3-D surfaces; however, by rendering merely the silhouette of the shape without lighting, LFD can only observe the condition of this shape on the edge of the silhouette [49]. D_A, which is the dissimilarity between two 3-D models, is calculated as

$$D_A\left(L_1, L_2\right) = \min_i \sum_{k=1}^{10} d\left(I_{1k}, I_{2k}\right), \quad i = 1, \ldots, 60 \tag{13}$$

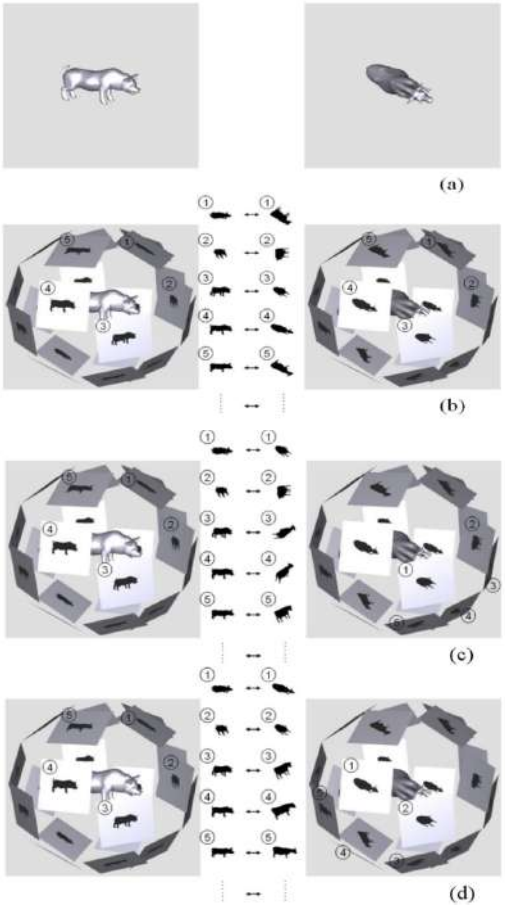


Fig. 5. *(a) Comparison of LFDs between two 3-D models: a pig and a cow. First, rendered images are extracted for both 3-D models. Then, as illustrated in (b), all 2-D images from the same views are compared, and a similarity value for this camera angle is obtained. Next, a different mapping between rendered images of the two 3-D models is chosen, and thus, another similarity value is extracted, as illustrated in (c). Eventually, the rotation of camera positions with the best similarity is found, as shown in (d). The similarity between the two 3-D models is attained by summing up the similarities from all the corresponding images [122].*

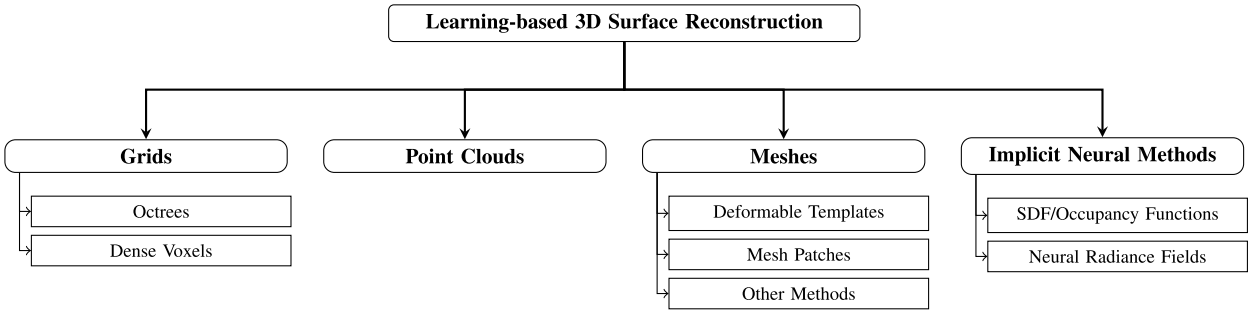


Fig. 6. *Taxonomy of learning-based reconstruction approaches based on 3-D shape representation.*

where i indicates different rotations between camera positions for two 3-D models, and I_1k and I_2k are corresponding images for the ith rotation. The dissimilarity between two images is denoted by d.

VI. DL-BASED 3-D SURFACE RECONSTRUCTION

DL-based 3-D surface reconstruction approaches can be broadly classified into four main categories according to their representation, as illustrated in Fig. 6:

1) Volumetric representations define a surface via small cuboids, either a dense 3-D voxel grid [20], [21], [22], [23], [24], [25] or an octree [26], [27], [28], [29]. Dense voxels are the 3-D analog of a pixel in 2-D space, i.e., a cubical element in a regularly spaced 3-D grid. Therein, octrees are obtained by recursively splitting 3-D space into octants, i.e., eight equally sized cells. In this data structure, only cells containing information by being close to the surface boundary are subdivided. Neighboring cells that have the same value do not need to be subdivided, and all of these areas can be represented by a single large octree cell. In order to achieve finer details, the space can be further partitioned into smaller octree cells, which is the main difference between a regular voxel grid and an octree.
2) Point-based representations utilize the constituting surface points to mark a shape [30], [31], [32], [33]. The entire surface is described through an unordered set of (x, y, z) coordinates.
3) Mesh-based representations describe an object through vertices, edges, and faces [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44]. Existing approaches can be mainly divided into three categories.
   a) Patch-based approaches attempt to reconstruct the final shape by learning a group of mappings from 2-D squares to 3-D patches and putting together these small patches.
   b) Deformable template-based approaches deform the vertices of a template mesh with predefined interconnections and predict the final shape based on it.
   c) Other mesh generation methods are so unique, yet singular, that they are sorted into a catch-all category.
4) Implicit neural representations describe a shape as a neural network that takes any (x, y, z) coordinate as input and maps it to occupancy or signed distance value [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56] or model radiance or appearance properties of an object such as NeRF-based approaches [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68].

Accordingly, we summarize and discuss the existing literature for DL-based 3-D surface reconstruction methods based on these categories in Sections VI-A–VI-D. Furthermore, we depict the architecture of different approaches with a unique color scheme in these sections. In the figures, data units are represented in red, trainable units in blue, and computing units in orange.

A. Volumetric Representations

Volumetric approaches in neural networks for 3-D surface reconstructions rely on describing the object through a grid. By extending the concept of 2-D convolutions to 3-D, a grid can be easily processed using learning-based approaches, such as neural networks.

Volumetric methods characterize 3-D object data using: 1) a regular 3-D voxel grid, i.e., dense voxels, and/or 2) an octree, i.e., sparse voxels.

Analogously to the concept of a pixel in the 2-D world, a voxel is a cubical element in a regularly spaced 3-D grid. An octree can be built by recursively subdividing the space into octants until a predefined maximum depth is reached. Additional information can be stored in cubic cells (both in dense and sparse voxels) to help reconstruct surfaces as follows.

1) Signed distance functions (SDFs) express the distance between the center of each voxel and the closest point on the surface of an object. They can be stored in a cuboid by calculating distance functions (DFs) [25], [123]. SDFs, a variation of DFs, purely calculate the signed distance value for each cell. Truncated SDFs (TSDFs) [124] go beyond the SDF definition by specifying a truncation threshold for SDF values stored in

cuboids, i.e., assigning a fixed value to voxels that are not near enough to the surface and their signed distance values exceed the defined threshold.

2) Occupancy or indicator functions indicate whether a cuboid is occupied by the surface of an object or not.

Learning voxel-based SDF representations is usually rather complicated compared to occupancy representations since dealing with DFs in 3-D space is more difficult than simply classifying a voxel as occupied or unoccupied [45]. However, voxel-based SDF approaches provide the advantage of generating smoother surfaces compared to occupancy grid-based approaches. A general disadvantage of voxel-based methods is their resolution limitation by the underlying 3-D grid. Mesh extraction approaches, such as the classical Marching Cubes (MC) algorithm [125], can be used to infer a mesh from the final output of these methods.

1) Dense Voxels: The majority of approaches with dense voxel-based representation voxelize the 3-D space in order to apply 3-D convolutional neural networks (CNNs) on a grid directly. In this section, we first present pioneer studies that applied CNNs to a 3-D representation, i.e., dense voxels, for shape classification and then introduce 3-D surface reconstruction and shape completion approaches that use dense voxels.

a) Volumetric CNNs for 3-D shape classification: Several studies have focused on solving shape classification and recognition tasks using dense voxels [21], [23], [98], [126], [127], [128], [129]. One of the pioneers in building DL models in 3-D world is 3-D ShapeNets, as proposed by Wu et al. [98]. They were among the first authors to show the application of CNNs to a 3-D representation. The introduced architecture uses a convolutional deep belief network for representing a 3-D shape as a probabilistic distribution of binary variables on a 3-D voxel grid. 3-D ShapeNet is able to conduct several tasks, from shape recognition to reconstruction and completion, as well as next-best-view prediction. The DL model takes a single-view depth map of the physical object as input and converts it into a volumetric representation. The occupancy status of each cell is specified by classifying it as either free space, unknown space, or observed surface. Next, a deep belief network is trained on this grid of size 30^3. In terms of accuracy, precision, and recall metrics, 3-D ShapeNets outperform several baseline methods for 3-D shape classification and retrieval, such as the LFD approach [122] and the spherical harmonic descriptor (SPH) [130], even though it utilizes a mesh at lower resolution. It was further shown that the DL model is able to automatically learn general 3-D features.

Maturana and Scherer [126] introduced VoxNet that voxelizes input point cloud data and processes the grid with a 3-D-CNN for object recognition tasks. The authors utilized a volumetric grid for representing the estimated spatial occupancy and a 3-D-CNN for extracting features and predicting class labels directly from the occupancy grid of size 32^3. Each point in the input point cloud is mapped to discrete volume coordinates. The resulting voxel volumes are fed to the proposed shallow neural network. VoxNet has fewer parameters compared to 3-D ShapeNets [98], i.e., less than one million versus over 12.4 million parameters, while achieving 8% and 6% higher average accuracy for ModelNet10 and ModelNet40 datasets, respectively. However, in both these methods, the memory and computational costs increase cubically with respect to the input resolution.

ORION [127], which is based on VoxNet [126], studies the importance of object orientation in 3-D object recognition results. Unlike VoxNet and 3-D ShapeNets [98], which augment training data with rotations of the objects to achieve rotational invariance of the network, ORION seeks to predict object orientation. The proposed network uses 3-D convolutional networks for 3-D recognition and adds an auxiliary orientation loss for better classification performance. By forcing the network to predict object orientation in addition to class labels during training, more accurate classification results can be achieved at test time. The ORION network is shallower than the proposed method by Brock et al. [21] that we discuss further down this survey, leading to fewer trainable parameters.

Some studies utilize multiview CNNs for analyzing a 3-D shape. Multiview CNNs work in three steps: 1) rendering a 3-D shape as a collection of images from different viewpoints; 2) inferring features for each viewpoint; 3) fusing these features across various views. In order to minimize the performance gap between multiview CNNs and volumetric CNNs, Qi et al. [128] suggested two new network architectures of volumetric CNNs. One architecture focuses on local regions, while the other uses anisotropic probing kernels for convolving a 3-D cube, then projecting 3-D volumes to a 2-D image, and afterward applying image-based CNNs for classification. The proposed CNNs surpass volumetric CNN-based methods, such as 3-D ShapeNets [98] and VoxNet [126]. Moreover, their classification accuracy competes with some multiview-based methods, such as MVCNN [131], LFD approach [122], and SPH [130], given the same 3-D resolution of 30^3.

b) 3-D surface reconstruction and shape completion using volumetric representation: In this section, we review the studies that leverage dense voxel representations for 3-D surface reconstruction [20], [21], [22], [23] and 3-D shape completion [24], [25]. Choy et al. [20] introduced a framework, 3-D recurrent reconstruction network (3D-R2N2), for both single- and multiview 3-D reconstructions. This method takes one or more RGB images of an object from arbitrary viewpoints as input and outputs a 3-D occupancy grid. The proposed network is composed of three main modules, as shown in Fig. 7: 1) a 2D-CNN, which encodes the input into a low-dimensional feature vector; 2) a 3-D convolutional long short-term memory (LSTM) [132], in which the 3D-LSTM units keep their previous cell states or update them, whenever there are more observations, i.e., multiview images, available; 3) a 3-D

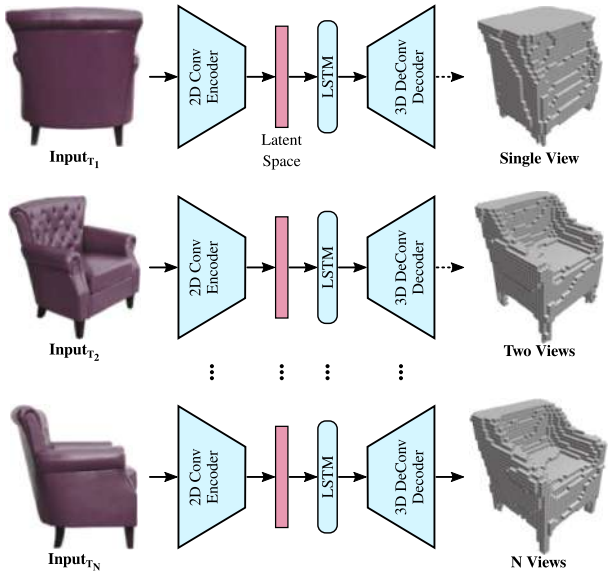


Fig. 7. *Overview of the 3D-R2N2 network [20]. The input to this network is one or more RGB images from arbitrary viewpoints, and the output is a 3-D occupancy grid. The main modules of 3D-R2N2 are an encoder, a 3-D LSTM, and a decoder.*

deconvolutional neural network (3D-DCNN) that decodes the 3D-LSTM hidden states into a higher resolution and produces the final occupancy grid.

In the LSTM module, 3D-LSTM units are located in a grid structure in such a way that each of them focuses on reconstructing a particular part of the output. Two versions of 3D-LSTMs, 3D-LSTMs without output gates and 3-D gated recurrent units (GRUs), were tried out in 3D-R2N2, in which the latter achieved better results. The output size is 32^3. Although the generation of detailed and thin parts of the objects and the reconstruction of objects with high texture levels are very challenging, 3D-R2N2 performs better than the category-specific approach proposed by Kar et al. [133], which learns 3-D shapes using camera viewpoint estimations together with object silhouettes, in SVR using real-world images. 3D-R2N2 is also able to produce accurate outputs compared to the MVS method [86] in multiview reconstruction (MVR).

Brock et al. [21] investigated generative and discriminative voxel modeling with deep ConvNet architectures. In short, their method presents a voxel-based variational autoencoder (VAE) [134], [135] for reconstruction and interpolation, a graphical user interface for investigating the latent space of autoencoders (AEs), and a deep voxel-based CNN for object classification. The output size of the network is 32^3. The voxel-based VAE learns to reconstruct features of an object, attaining acceptable reconstruction accuracy. It further facilitates the transition from one object to another by interpolating between their reconstructions. The neural model has significantly fewer parameters than FusionNet [129], i.e., 18 million as opposed to 118 million. Nevertheless, it achieves competitive results compared to ORION [127] considering that ORION uses orientation augmentations to improve the classification.

The TL-embedding network [22] learns a vector representation of an object, which is both generative in 3-D, i.e., able to reconstruct objects in 3-D space from this representation, and predictable from 2-D images, i.e., able to extract this representation from images. As shown in Fig. 8, this architecture is composed of a convolutional network, which brings about predictability, and an autoencoder, which results in generativeness. It generates outputs with 20^3 resolution. This method captures stylistic details better than the method proposed by Kar et al. [133].

Wu et al. [23] introduced a framework, called 3-D generative adversarial network (3D-GAN), which generates novel volumetric 3-D objects from a probabilistic latent space. 3D-VAE-GAN, an extension of 3D-GAN, provides the ability to reconstruct surfaces from input images. For generation and recognition of 3-D objects, this method utilizes both general-adversarial modeling [136], [137] and volumetric convolutional networks [98], [126], as illustrated in Fig. 9. Furthermore, it fuses 3D-GAN with a VAE [134] for 3-D object reconstruction from a single 2-D image. The resolution of its final output can reach up to 64^3. The classification accuracy of this network is roughly similar to volumetric learning-based approaches, such as VoxNet [126] and ORION [127], but is lower than the method proposed by Qi et al. [128]. It shows higher AP for voxel prediction compared to the work by Girdhar et al. [22] in a single-image 3-D reconstruction task. However, 3D-VAE-GAN usually creates a noisy and incomplete output from an input image. Studies conducted by Wu et al. [138] showed that, ultimately, training GANs together with recognition networks can lead to high instability.

Stutz and Geiger [25] introduced a learning-based approach with weak supervision for 3-D shape completion. It takes a 3-D bounding box and an incomplete point cloud as input and predicts the complete object shape. The completion process is done in two steps. 1) A shape prior is learned, i.e., a VAE is employed to learn a 3-D

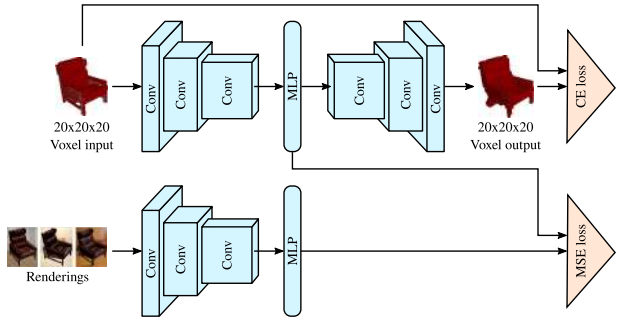


Fig. 8. *TL-embedding network [22]. During training, two types of input are fed to the network: 2-D RGB images as the input to ConvNet at the bottom and 3-D voxel maps as the input to the autoencoder at the top. The network outputs a 3-D voxel map.*

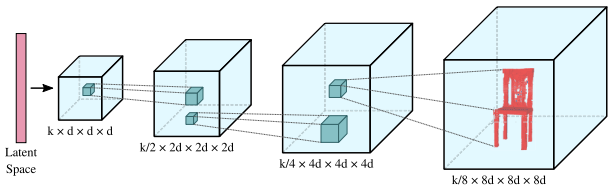


Fig. 9. *Generator architecture in 3D-GAN [23]. Five volumetric fully convolutional layers of kernel sizes 4 × 4 × 4 and strides 2 make up the generator. The discriminator architecture is usually mirroring the generator architecture.*

shape model on synthetic data, encoding shape models in a dataset using occupancy grids and SDFs at 24 × 52 × 24 resolution. 2) Shape inference is performed. For this, 3-D shape completion is considered a maximum likelihood (ML) problem. The authors used the amortized ML (AML) approach that works over the lower dimensional latent space z from the first step. It keeps the pretrained decoder from the previous step fixed and adds a new encoder. The encoder is trained without supervision, i.e., without using explicit labels, and learns to directly predict ML solutions from incomplete input observations using ML loss. The presented method was shown to be faster than a fully supervised baseline while using 9% or less supervision while being able to produce competitive results.

Dai et al. [24] fused a volumetric DNN with a 3-D shape synthesis procedure to complete partial 3-D inputs. Their approach generates the output in two major stages. 1) A shape prediction method, which predicts a volumetric grid with 32^3 resolution as a low-resolution global structure of the input. The proposed network, the 3-D-encoder-predictor network (3D-EPN), consists of 3-D convolutional layers and attempts to predict distance field values for missing data. 2) A patch-based 3-D shape synthesis method, which employs a synthesis procedure to improve local details and create a high-resolution output using CAD model priors. Given the predicted coarse output from the first stage, the authors carried out a search for similar 3-D shape models in the ShapeNet [96] database. Based on the results, they sought to find similar local patches in these shape models for the purpose of local detail synthesis. The resolution of the final voxel grid is 128^3. Without the synthesis step, 3D-EPN provides only low resolution and is unable to predict local details and fine structures. Nevertheless, it outperforms 3-D ShapeNets [98] and Poisson methods [8], [9].

In another approach, Dai et al. [139] suggested sparse generative neural networks (SG-NNs), which is a self-supervised scene completion approach that accepts an incomplete RGB-D scan as input and predicts a high-resolution 3-D reconstruction while also inferring unseen, missing geometry. The self-supervised nature of this technique allows for training entirely on real-world, partial scans. This eliminates the requirement for synthetic ground truth. Self-supervision is achieved by removing some frames from a given (incomplete) RGB-D scan, resulting in an even more incomplete input; this input is used to create an input-target pair (the original scan is considered the target scan). The difference in partialness is then correlated in this input-target pair, while regions that have never been observed are masked out during training. Despite the fact that fully complete scenes are not used as samples during training, this approach generates high levels of completeness by learning to generalize completion patterns across the training set. Dai et al. also proposed an SG-NN, a fully convolutional encoder–decoder architecture, capable of predicting high-resolution final geometry as a sparse TSDF representation. This end-to-end formulation generates a 3-D scene in a coarse-to-fine manner. SG-NN is built upon sparse convolutions [140] that operate only on surface geometry. This self-supervised approach produces more accurate and complete scenes in comparison to a fully supervised approach, such as 3D-EPN [24].

In general, voxel-based methods encounter a number of difficulties. Information loss may occur due to discretization and transformation of input data to coarse voxels. Moreover, cubic growth in memory limits the resolution and the overall computational demands bring about coarse final outputs. Generating higher resolution surfaces requires deeper networks. However, the network depth is constrained by the available GPU memory. Therefore, it may affect the ability of CNNs with volumetric decoders in producing high-resolution outputs [141].

2) Octrees: Dense voxel representations are associated with a number of challenges regarding resolution, memory, and computational complexity. In many cases though, the 3-D shape surface occupies only a small portion of 3-D space. Hence, octrees mark a popular approach for partitioning space, as they allow for the 3-D data to be stored in a sparse structure [142], [143]. For octree construction of a 3-D shape, a bounding cube is created around the entire shape. This bounding cube will be recursively subdivided. In each step, all cuboids, which are occupied by a shape boundary, are traversed, and each of them is divided into eight smaller, equally sized cuboids. However, in order to enable CNN operations on an octree, this data structure needs to be updated and slightly changed, which leads to complex implementations, while the resolution is still limited by the underlying 3-D grid [45]. Hence, convolutions and pooling to octrees are applied similar to CNN operations on dense voxels with the main difference being that the elementary operand is an octant.

a) OctNet: Riegler et al. [144] presented OctNet that enables the usage of high-resolution inputs for DL purposes. OctNet is based on a 3-D-CNN that can be applied to a special form of octree data structure to learn representations from high-resolution 3-D data. Vanilla octree implementations might encounter data access speed issues in high-resolution (high recursion depths) octrees. On the other hand, for convolutional network operations, such

as convolution or pooling, it is crucial to have frequent access to different data elements, such as cell neighbors. In order to provide faster data access and reduce cell traversal time, the authors proposed a hybrid grid-octree data structure. They used a shallow octree, which is an octree with maximum depth $D = 3$, as a basic building block. Several of these shallow octrees are stacked in a regular grid structure to cover the whole volume. Input resolution effects of this representation were evaluated on three different tasks: 3-D classification, 3-D orientation estimation of unknown object instances, and semantic segmentation of 3-D point clouds. For high-resolution inputs in the 3-D shape classification task, OctNet runs faster and requires less memory as opposed to DenseNet, a densely voxelized version of OctNet. In general, both OctNet and DenseNet perform better than a shallow network such as VoxNet [126], verifying that network depth is of great importance.

OctNet does not generate an octree structure, and this structure has to be known in advance for both input and output. In classification and semantic segmentation tasks, this does not comprise a problem. However, learning the volumetric structure of objects and scenes, and being able to construct them is crucial in generative tasks, such as reconstruction, generation, and completion, since the input and output partitioning structure might be different. OctNetFusion [26] proposes a learning-based approach, which learns to partition the space and can predict an SDF or a binary occupancy map. The network takes one or more 2.5-D depth maps as input. To reconstruct precise and complete 3-D outputs, it fuses depth information from different viewpoints into a coarse volumetric grid. Then, this volumetric grid (grid-octree structure) is fed to the OctNetFusion network architecture, consisting of encoder–decoder modules. The network determines whether a cell should be subdivided or not in a coarse-to-fine manner. The output resolution can be up to 256^3. This approach performs qualitatively and quantitatively better than traditional volumetric fusion approaches, such as vanilla TSDF fusion [124] and TV-L1 fusion [145] for volumetric fusion tasks and Voxlets [146] for volumetric shape completion from a single image.

b) O-CNN: Another concurrent work in the scope of octree-based CNNs (O-CNNs) for 3-D shape analysis is the O-CNN [147]. The authors' main idea is to represent 3-D objects with octrees and execute 3-D-CNNs only on nodes or cuboids, which are occupied by boundaries of the 3-D object, instead of sliding the convolutional kernel over the whole voxel grid, as done for the standard convolution computation in full voxel grids. The network constructs an octree from an input-oriented 3-D model, e.g., an oriented triangle mesh or a point cloud with oriented normals, and enriches each octant of this data structure with metainformation, such as shuffle key vectors, label vectors, and input signal, which are needed for the convolution operations. Furthermore, a hash table is built to accelerate neighborhood search in the convolution. By storing the octree data structure in the graphical memory, O-CNN can be easily and efficiently trained and evaluated on GPUs. To demonstrate the efficiency of their network, the authors evaluated it on three shape analysis tasks: object classification, shape retrieval, and shape segmentation. In terms of classification accuracy, O-CNN performed better than VoxNet [126], slightly worse than the method proposed by Brock et al. [21], and competitive to nonvoxel-based methods, such as PointNet [148]. In addition, the impact of different input representations on the same network architecture (O-CNN) was investigated. Results showed that an octree input achieves higher accuracy compared to full voxel structures. For object part segmentation, O-CNN yields better or comparable performance than other methods, such as PointNet [148].

To improve the computation and memory efficiency of O-CNN, Wang et al. [27] proposed the extension "Adaptive O-CNN," which consists of an encoder–decoder structure and uses patch-guided adaptive octree shape representations. Contrary to approaches such as volumetric-based CNNs, where the output is generated as voxels of the same resolution, this method can generate adaptive octrees based on a patch-guided partitioning strategy and with differently sized planar patches. The underlying assumption is the subdivision rule, which states that splitting all octants to the finest level is not necessary. The process can be stopped early for some of the octants, and the local shape inside these octants can be represented by simple patches, e.g., planar patches. However, this approach limits the quality of the output and may encounter some difficulties in generating watertight and curved surfaces. Adaptive O-CNN obtains better or comparable classification accuracy than PointNet [148], OctNet [144], and O-CNN [147], yet it performs worse than PointNet++ [149], Kd-Network [150], and the method proposed by Brock et al. [21]. For the task of shape reconstruction from a single image, Adaptive O-CNN surpasses PointSetGen (PSG) [30] and AtlasNet [34] in generating more detailed geometry.

c) Other octree prediction approaches: Häne et al. [28] introduced a hierarchical surface prediction (HSP) framework for high-resolution voxel grid prediction in 3-D object reconstruction. The main idea boils down to generating and predicting high-resolution voxels around the predicted surface and coarse-resolution voxels for the interior and exterior parts of an object. The high-resolution voxels are not predicted directly, but, instead, a coarse-to-fine approach is used to create smoother 3-D models hierarchically and in a multiresolution fashion. Starting with approximating the coarse geometry of the output, more finely resolved details are added step by step by refining the surface. This process, finally, results in a voxel grid with up to 256^3 resolution. The proposed method is based on an encoder–decoder architecture. A convolutional encoder encodes input to a feature vector, and then, an upconvolutional decoder predicts the voxel grid or final data structure (called voxel block octree

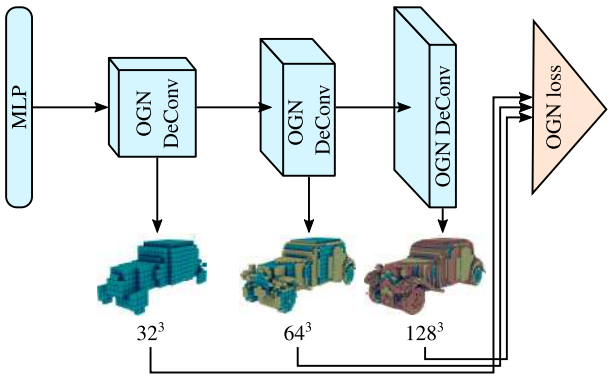


Fig. 10. *OGN [29] takes an input 3-D shape and gradually reconstructs octrees as the output in different resolutions.*

data structure in this article). Classifying each voxel as the boundary, free space, or occupied space, only voxels with a boundary label require high-resolution prediction since they cover the actual surface. The major difference between HSP and OctNet [144] is that OctNet takes the structure of the shallow octrees as input, while HSP predicts the structure of the tree together with its content. HSP produces more accurate surfaces with higher resolutions compared to low-resolution baselines predicting dense voxels.

In a similar approach, Tatarchenko et al. [29] suggested an octree generating network (OGN) that is a convolutional decoder that can generate and predict the octree structure of 3-D shapes, along with the occupancy value of each cell. It operates on octrees and reconstructs 3-D shapes in a multiresolution manner, as illustrated in Fig. 10. This method generates results up to a resolution of 512^3. The network gradually reconstructs a high-resolution surface from the initial, low-resolution dense voxel grid using hash-table-based octree blocks. If the reconstructed surface has not yet reached the final output resolution, cells with a "mixed" state, i.e., undetermined state, will be passed to the next layer of the network for further subdivision. Providing the same accuracy as dense voxel grids in low resolutions, OGN offers less memory consumption and shorter run-time in higher resolutions in comparison to voxel grid-based networks. In particular, it is 20 times faster and requires two orders of magnitude lower memory usage at 512^3 resolution.

B. Point-Based Representations

These days, point clouds are becoming increasingly important and available due to the improvements in scanning devices in recent years. A point cloud is a set of points in 3-D space, inferred by various 3-D data acquisition techniques. It is an irregular data format since there is no canonical order between the points in a set. Each point can be defined by its (x, y, z) coordinates. Therefore, the size of the matrix representing a 3-D object is initially $N \times 3$ for N points. The number of columns in this matrix representing the features might be extended if other information, such as color and normal, exists. Considering the irregular and unordered nature of point clouds, it is difficult to apply DL techniques, such as CNNs, directly on them. Consequently, in order to process a point cloud with neural networks, it was common to transform them into voxel grids or collections of images. These transformations usually present numerous challenges, such as information loss, voluminous data, resolution constraints, and high computational costs. To reduce the overhead of data transformation to other data formats, different methods for effectively processing point clouds with neural networks have been proposed, which will be discussed in Sections VI-B1 and VI-B2.

1) PointNet and PointNet++: Pioneer works in the field of learning global features directly on point clouds are PointNet [148] and PointNet++ [149]. PoinNet as proposed by Qi et al. [148] directly consumes a raw point cloud as an input and uses it for discriminative DL tasks, e.g., object classification, semantic segmentation, and part segmentation. As illustrated in Fig. 11, each of the points in the input set is processed by a small neural network individually and independently based on its own coordinates, resulting in a high-dimensional embedding of the points. Following the embedding step, a simple symmetric function, such as max pooling, is utilized to aggregate the encodings from each of the points. The symmetric function is chosen such that it pays attention to the permutation invariance of the input points. The aggregation step brings about a global feature vector, which encodes the whole shape and can be fed to different neural networks for recognition purposes. PointNet achieves higher classification accuracy compared to the LFD approach [122], which is a 3-D model retrieval method, SPH [130], and other methods with volumetric representation, such as 3-D ShapeNets [98], VoxNet [126], and another method previously proposed by Qi et al. [128]. Although it has around 17 times fewer parameters than multiview-based methods, such as MVCNN [131], its performance is only slightly lower compared to these methods. PointNet pro-

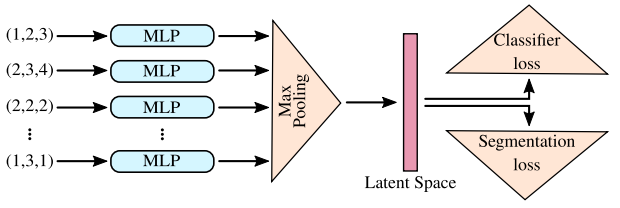


Fig. 11. *PointNet architecture [148], which is used for classification and segmentation tasks, directly accepts a point cloud as input. Each of the points in the input point cloud is processed by a small neural network individually and independently. Then, point features are aggregated by max pooling, a simple symmetric function that respects the permutation invariance of the input points. The aggregation step creates a global feature vector that encodes the entire shape.*

vides linear complexity $O(N)$ in both spatial and temporal domains, where N is the number of input points, while the complexity grows squarely with respect to image resolution for multiview methods and cubically with respect to the volume size for volumetric methods. More importantly, due to it satisfying the permutation invariance condition, PointNet cannot capture local information and, thus, lacks generalization.

In order to resolve the issues of PointNet, Qi et al. [149] introduced the extension PointNet++, which pays more attention to local features and combines them with global features to infer better results. The architecture is built on top of PointNet, enriching it with a hierarchical feature learning approach. The whole process, which is done recursively, can be summarized as follows: 1) specifying centroids of local regions by sampling a subset of the input point cloud using the farthest point sampling (FPS) algorithm; 2) finding local neighborhoods of these centroid points using radius-based ball query; 3) applying a mini-PointNet in each neighborhood to mimic the concept of a convolution kernel and conduct convolution-like operations in point space for the purpose of local feature extraction. The presented method proved to be robust toward nonuniform sampling density, which might occur due to perspective effects, variations in radial density, motion, and so on. Compared to PointNet, PointNet++ has an improved classification accuracy for the ModelNet40 dataset.

2) Point cloud reconstruction and generation: PointNet was mainly implemented for discriminative tasks, such as classification and segmentation. The first approach for reconstructing a 3-D point cloud of an object from a single (monocular) RGB or RGBD image was proposed by Fan et al. [30] and is based on a generative learning-based approach. The main contributions of this work are given as follows: 1) designing a point set generator network; 2) proposing two proper loss functions for the comparison of the ground truth with the network's predictions for point sets, i.e., CD and EMD; 3) modeling uncertainty and ambiguity of the ground truth. The proposed network is composed of an encoder and a predictor part. The encoder transforms the input into an embedding space. The predictor is divided into two parallel branches: a deconvolution (deconv) branch and a fully connected (fc) branch. The deconvolution branch learns the smooth parts and main body of the object, while the fc branch learns nonsmooth parts and details. The results of these branches are then concatenated to create the final point set. In comparison to 3D-R2N2 [20], which generates a volumetric representation from single or multiview images, this method produces better results on CD, EMD, and IoU metrics. In addition, it is able to reconstruct thin structures more accurately.

Achlioptas et al. [31] proposed a solution for generative tasks and unsupervised representation learning based on an end-to-end pipeline that can reconstruct point clouds using deep autoencoders (AEs) and GANs. The autoencoder extracts features by learning a lower dimensional representation of the input, based on which the GAN [136] generates point clouds. In the autoencoder architecture, the authors exploited a PointNet-like encoding scheme to learn compact representations. The encoder generates a latent code that is invariant to the order of input points. The latent code is converted back to a point cloud using a standard deep network with three fc layers as a decoder. The authors further investigated three different approaches for point cloud generation: 1) GAN operating on raw point cloud; 2) latent-GAN, which is a plain GAN being trained on the latent space of the pretrained AE; 3) Gaussian mixture models operating on the latent space learned by AE. The study indicated that the proposed AE provides good generalization capacity toward unseen data. However, the output of the proposed DL model architecture is limited to 2048 points, and generating high-quality surfaces with such a small number of points is challenging.

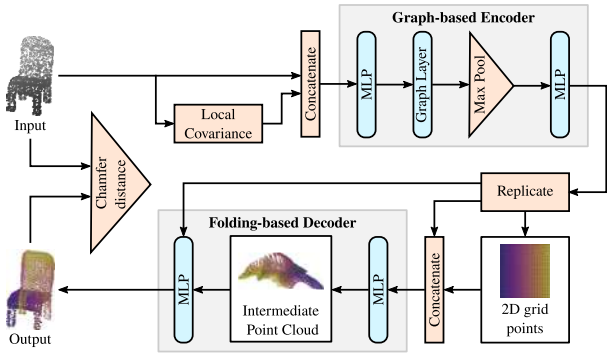


Fig. 12. ***FoldingNet architecture [32] consists of a graph-based encoder (an improved and generalized version of PointNet), which encodes local neighborhood structure information, and a folding-based decoder, which reconstructs the point cloud from a 2-D grid template deformation process.***

Another closely related approach that attempts to solve unsupervised learning challenges using deep autoencoders is FoldingNet [32]. The presented architecture, as illustrated in Fig. 12, utilizes a simple graph-based scheme as the encoder part (similar to the method proposed in [151], an improved and generalized version of PointNet) in order to encode local neighborhood structure information. Since applying convolution operations on graphs is difficult, the authors suggested building the k-nearest neighborhood graph (K-NNG) and repeatedly applying max-pooling operations on each node's neighborhood. This way, the DL model is able to capture locality and extract features of neighboring points. For the decoder part, a folding-based scheme is proposed to reconstruct the point cloud from a 2-D grid template deformation process. Due to the fact that 3-D point clouds are often sampled from object surfaces, one can make the assumption that any 3-D object surface can be converted and squeezed into a 2-D plane. It is also possible to reverse this process, i.e., wrapping 3-D shapes with a fixed 2-D paper (plane). This property builds the foundation of the proposed method.

The decoder maps 2-D points from a 2-D template grid to the surface of the 3-D object using folding operations. The definition of the folding operations, i.e., 2-D-to-3-D mapping, is the main contribution of this article, making it the first single learned parametric function embedding from a (gridded) 2-D (point) manifold into 3-D space and a fundamental building block for other surface reconstruction approaches. FoldingNet's decoder requires about 7% of the parameters of the fc decoder proposed by Achlioptas et al. [31], which is significantly smaller than the latter. However, it was shown to perform better at feature extraction in terms of classification accuracy and reconstruction loss. Overall, FoldingNet achieves higher classification accuracy than other unsupervised methods, such as LFD approach [122], SPH [130], TL-embedding network [22], and 3D-GAN [23].

PointFlow [33] is a 3-D point cloud generation framework that learns a distribution of distributions, i.e., the distribution of shapes and its respective points. A VAE is applied to transform sampled 3-D points from the point prior into a realistic point cloud conditioned on a shape vector. The distributions are modeled in two steps. First, the distribution of the latent space of shapes is learned. To enable the method to sample multiple shapes, PointFlow extracts latent vectors of different shapes. A sampled Gaussian vector (a shape prior) is transformed into a shape latent vector using a continuous normalizing flow (CNF) [152], [153], [154]. In the second step, the distribution of points on a specific shape is learned for shape generation. Given a sampled 3-D Gaussian point cloud (point prior) and a shape latent vector inferred from the first step, a CNF is used to move input points to their new location and transform them into the target shape. For generative tasks, PointFlow outperforms the methods proposed in [31] in terms of the 1-nearest neighbor accuracy (1-NNA) metric while having fewer parameters. With respect to the EMD score, it achieves better autoencoding performance compared to Achlioptas' method [31] for point cloud reconstruction from inputs.

Several recent studies have investigated point cloud upsampling [155], [156], [157], normal and curvature estimation from point clouds [158], [159], classification [160], [161], [162], [163], [164], [165], [166], segmentation tasks [160], [161], [162], [164], [166], [167], [168], object detection [163], [169], and point cloud denoising [170]. Although point-based representation approaches discretize the surface of the shape into a set of 3-D points, they do not model the corresponding connectivity. Thus, additional postprocessing steps are needed to generate the final high-quality 3-D mesh. On the other hand, existing approaches are very limited in terms of the number of generated points leading to limited output quality.

C. Mesh-Based Representations

Meshes are irregular types of data that are difficult to predict by neural networks. Their components are vertices, edges, i.e., pairs of vertices, and (triangular) faces, i.e., triplets of vertices. Therefore, researchers have investigated different paths to address mesh-based representations, namely, patch-based approaches [34], [37], deformable template-based approaches [38], [39], [40], [41], [70], [171], and other mesh generation methods [35], [36], [42], [43], [44], [172].

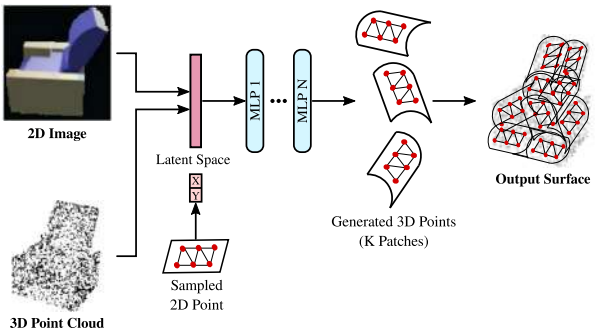


Fig. 13. *AtlasNet [34] is a patch-based approach that takes either a 2-D image or a 3-D point cloud as input and outputs a 3-D mesh. MLPs are used to estimate the target 3-D surface, which learns the local mapping of 2-D-points to 3-D-surface points.*

1) Patch-Based Approaches: Groueix et al. [34] introduced a method for 3-D surface generation, called AtlasNet, as illustrated in Fig. 13. They suggested generating a 3-D surface and representing it as a set of folded 2-D squares. The input shape can be either a 2-D image or a 3-D point cloud. The method outputs the corresponding 3-D mesh and its atlas parameterization. The main steps of the approach include encoding an input 3-D point cloud into a 3-D shape and reconstructing the 3-D shape from an input RGB image. 3-D point clouds are encoded using a PointNet-based encoder, which transforms the input point cloud into a 1024-D latent vector. Input images are encoded using ResNet-18 [173]. The decoder consists of four fc layers, which extract the final surface. The target 3-D surface is estimated using multilayer perceptrons (MLPs), which learns the local mapping of 2-D-points to 3-D-surface points. Therefore, by transforming the 2-D squares to the 3-D surface using learnable parametrizations, i.e., MLPs or patches, the final surface is covered in a way similar to putting paper strips on a shape to make a papier-mâché. The difference between the proposed method and FoldingNet [32], which is a folding-based method, is that FoldingNet deforms just one 2-D square or patch, while AtlasNet investigates a varying number of 2-D squares. Results from AtlasNet showed that the usage of multiple patches improves 3-D reconstruction. For SVR from a 2-D RGB image, AtlasNet yields qualitatively better performance compared to the dense voxel-based method 3D-R2N2 [20], the octree-based method HSP [28], and a point-based method [30]. Furthermore, it was shown that AtlasNet provides good generalization properties;

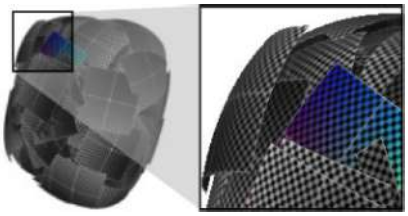


Fig. 14. *Meshlet inconsistencies adapted from the patch-based approach paper [37].*

however, it generates artifacts such as self-intersecting parts and overlapping patches.

Badki et al. [37] proposed an approach to extract a 3-D mesh from a noisy, sparse, unordered, and nonoriented set of points. Instead of learning shape priors at the object level, the method learns them locally while enforcing global consistency. In order to represent these priors and local features, small mesh patches, called meshlets, were used. These meshlets can be interpreted as a dictionary of local features and learned priors. The final mesh is the union of all meshlets. The authors used a VAE for learning the priors by using a very large dataset of meshlets, which was extracted from objects in the ShapeNet dataset. During training, the local priors are learned with meshlets. At inference, meshlets are deformed to match the input point cloud via distance minimization. Since individual meshlets are updated independently in order to adapt to the points, the overall mesh extracted from their union is not watertight. Therefore, a global consistency step is performed to eliminate inconsistencies across all meshlets, as illustrated in Fig. 14. Compared to occupancy networks [45] and AtlasNet [34], which are class-specific algorithms that learn priors at the object level, and deep geometric priors [174], this method produces better quantitative results in terms of CD and HD metrics. It also performs qualitatively well at reconstructing objects from unseen classes during training, coping with noise, and being robust to dramatic changes in the object's pose.

For all the aforementioned methods, mesh patches and the tessellation process may affect the quality of the final surface, especially for complex shapes. Therefore, these approaches may generate self-intersecting meshes and might be unable to generate closed surfaces.

2) Deformable Template-Based Approaches: Deformable template-based approaches take a template mesh with predefined interconnections as input, deform the vertices, and predict the final shape based on this. These approaches can generally reconstruct meshes and shapes with simple topology; however, they struggle to generate complex structures with a lot of details. Wang et al. [38] designed Pixel2Mesh, an end-to-end reconstruction pipeline for extracting a 3-D triangular mesh from a single RGB image. Taking an input image and an ellipsoid with fixed numbers of edges and vertices as the initial mesh, it gradually deforms the mesh using a graph-based CNN [graph convolutional network (GCN)] to generate the final 3-D shape. As illustrated in Fig. 15, the overall method is composed of two main parts. 1) An image feature network (2D-CNN), which is used to infer perceptual features using an input color image. 2) A three-block cascaded mesh deformation network (graph-based ResNet) that takes care of initial mesh deformation in a coarse-to-fine manner. Each graph-based ResNet block takes the perceptual feature concatenated with 3-D feature encoding of the input mesh as input. In their study, the authors showed that Pixel2Mesh outperforms 3D-R2N2 [20] and the point-based method proposed by Fan et al. [30] in terms of the mean of F-score, CD, and EMD metrics. Qualitywise, it produces smoother surfaces with local details. Nevertheless, the approach shows generalization issues and can only generate meshes and objects of topologies similar to the initial mesh.

Pixel2Mesh++ [39] works along with Pixel2Mesh to produce 3-D meshes from multiview images. The main idea is that adding more images (three to five) of an object as input provides more information for a shape generation method and, thus, results in more accurate and detailed reconstructions. Pixel2Mesh++ consists of a multiview deformation network (MDN), which processes cross-view information for the prediction of optimal deformations. First, a coarse mesh is produced by Pixel2Mesh, which is then fed to the MDN part to be refined progressively by adding details. With regard to the F-score metric, Pixel2Mesh++ generates better results than 3D-R2N2 [20], learned stereo machine (LSM) [175], and two other baselines that the authors implemented using Pixel2mesh [38]. In addition, it generalizes well across various semantic categories and produces high-quality outputs with accurate details.

Recent efforts by Kanazawa et al. [40] utilized a CNN image encoder followed by three modules for 3-D shape generation, camera pose estimation, and texture prediction. The CNN acts as an encoder, producing a latent representation of a single input image, which is fed to the three prediction modules. The 3-D structure of a shape is generated by deforming a learned category-specific mean shape with instance-specific predicted deformations. Texture is parameterized as a UV image that is predicted using texture flow. This mechanism enables the method to transfer the texture of one instance onto another. However, it cannot produce the detailed structure of the input shape. The presented approach obtains comparable results to the one proposed by Kar et al. [133] in terms of the IoU metric. Kar et al. [133] exploited segmentation masks and optionally a set of keypoints as annotations during inference to generate 3-D rigid objects. Contrary to that, the method of Kanazawa et al. [40] only utilizes these annotations during training and directly predicts a 3-D structure form an unannotated input image at inference time.

Hanocka et al. [41] introduced Point2Mesh for reconstructing meshes from point clouds. The core idea is a mesh fitting process for the reconstruction of the final mesh.

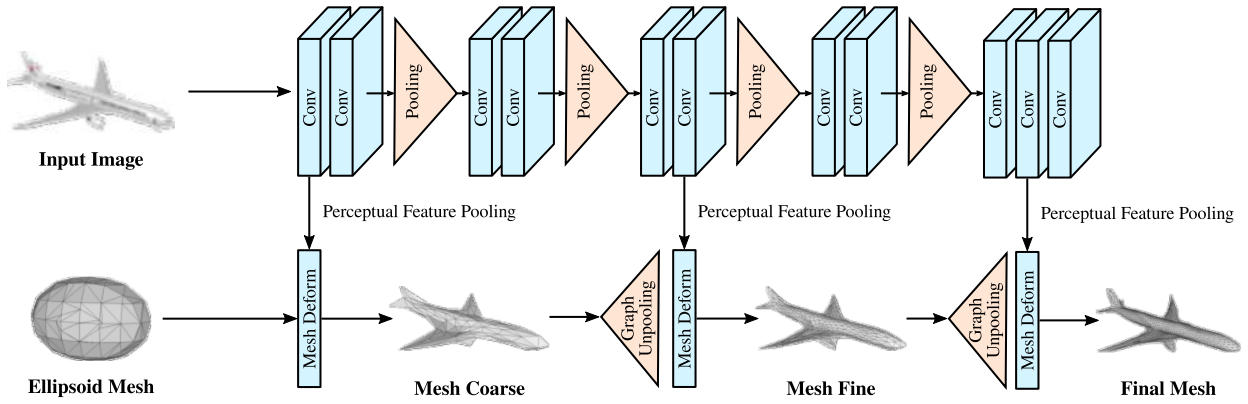


Fig. 15. *Pixel2Mesh network [38] is a deformable template-based approach that reconstructs a 3-D triangular mesh from a single RGB input image. It consists of three mesh deformation blocks used for mesh resolution enhancement and vertex location estimation.*

In addition to the input point cloud, an initial watertight mesh is fed to the network. This initial mesh represents a coarse approximation of the point cloud, which is iteratively deformed from outside-in using a CNN to fit the input point cloud, as illustrated in Fig. 16. Accordingly, a network learns displacement and deformation of the mesh vertex positions. The optimization of Point2Mesh is based on MeshCNN [176], which is a CNN-based pipeline applied on triangular meshes. Unlike Screened PSR, Point2Mesh is agnostic to normal orientation and ensures watertight reconstructions from noisy input with missing parts and unoriented normals. It also achieves a higher F-score compared to Screened PSR [9] and deep geometric priors [174] for shape denoising and completion. However, Point2Mesh requires a large amount of compute time and memory, possibly alleviated by data parallelism or model parallelism [177].

3) Other Mesh Generation Methods: Liao et al. [42] investigated end-to-end 3-D surface prediction using a differentiable MC (DMC) algorithm. In previous research, the surface prediction was solved in two steps: first, predicting an intermediate SDF/occupancy representation using an auxiliary loss, and second, taking a postprocessing step for 3-D mesh extraction separately, such as the MC algorithm. On the other hand, applying backpropagation to the MC algorithm is intractable due to nondifferentiability. Hence, in order to unite these steps to create an end-to-end framework, the authors inserted a differentiable formulation as a final layer into a 3-D-CNN. A point cloud, which is used as input, is directly converted into a volumetric representation using a grid pooling operation, e.g., max pooling in each cell. An encoder–decoder network with skip connections is then used to process pooled features, with the decoder operating in volumetric space. That way, it not only estimates occupancy probabilities but also predicts the vertex displacement field for a surface mesh. Compared with baseline methods that infer occupancy or TSDF first and then apply MC as a postprocessing step, DMC achieves superior results with respect to CD, accuracy, and completeness metrics. Nevertheless, difficulties may arise while reconstructing very thin surfaces, and disconnected parts can become connected.

Fig. 16. *Point2Mesh [41] takes a point cloud (in blue) and a deformable initial mesh as input and gradually reconstructs the final output shape.*

Scan2Mesh [43] is a generative model that combines convolutional and graph neural network architectures to predict a complete, lightweight, and structured 3-D mesh representation from an unstructured and incomplete range scan of an object. The aim is to predict both vertex location and edge. Initially, the feature space is computed through a set of 3-D convolutions from input TSDF. The vertices are then predicted based on the extracted features. An fc graph is generated from the predicted vertices, and all of the vertices are connected to each other via edges. Next, a graph neural network is used to classify edges and extract the ones that belong to the mesh graph structure. Using this intermediate graph of predicted edges and vertices, a dual graph is created which comprises a set of valid potential faces. Finally, another GNN is applied to predict the final face structures from the dual graph. Scan2Mesh offers better qualitative and quantitative performance compared to 3-D ShapeNets [98], 3D-EPN [24], and PSR [8], [9]. However, it depends on fc graphs for predicting edges, which leads to limitations in model size (MS).

Mesh R-CNN [44] is an approach that unifies both 2-D perception and 3-D shape prediction. It takes a single RGB image as an input, detects 2-D object instances in the image, and creates a category label, bounding box, segmentation mask, and 3-D mesh predictions of the detected objects as the outputs. Mesh R-CNN utilizes

Mask R-CNN [178], an end-to-end region-based 2-D object detector, for the detection of 2-D objects. The 3-D shape prediction step depicted in Fig. 17 is based on a hybrid approach, which primarily produces a coarse voxel representation of a detected object, transforms this voxelization into an initial 3-D triangular mesh, and, finally, refines this mesh by modifying the vertex positions using a GCN. This approach achieves better results compared to a voxel-based method, such as 3D-R2N2 [20], a point-based method [30], and a mesh-based method, such as Pixel2mesh [38] in single-image shape prediction considering CD and F1-score metrics.

Liu et al. [35] attempted mesh reconstruction from input point clouds by fully utilizing the input and simply adding connectivity to the existing points. Toward this end, they introduced a deep point cloud network that proposes candidate triangles and predicts faces. This information is provided as input to a mesh generation module. First, a k-nearest neighbor (k-NN) graph is built for each point in the input point cloud, in order to decide which three points should form a triangle face and infer candidate triangle proposals. Next, an MLP network is employed to classify candidate triangles and filter out incorrect triangles, such as the ones that connect two independent but spatially adjacent parts of the shape, using the intrinsic–extrinsic ratio (IER). To infer the local connectivity between vertices comprising a triangle, the ratio of the geodesic distance (intrinsic metric) and the Euclidean distance (extrinsic metric) was proposed. Finally, in a postprocessing step, the remaining candidate triangles are sorted and merged in a greedy way to generate the final mesh. The approach outperforms several learning-based methods, such as AtlasNet [34], deep geometric priors [174], deep MC [42], and DeepSDF [50], as well as traditional reconstruction methods, such as PSR [8], [9], MC [125], and BPA [10] in terms of F-score, CD, and NC metrics. Moreover, it generates higher quality outputs with fine-grained structures than the aforementioned methods and offers the capability to be transferred to unseen categories.

Daroya et al. [36] proposed a recurrent neural network (RNN)-based method, called recurrent edge inference network (REIN), to produce triangulated surface meshes from sparse input point clouds using a bottom-up approach. The network tries to predict edges sequentially and generates a mesh by processing points one at a time from a queue of points. The latent vector of the input point cloud, which is inferred by a PointNet-based [148] autoencoder, is also used to enrich the data with global structure information of an object. For edge prediction, the authors relied on the application of recurrent networks, inspired by GraphRNN [179]. An RNN can be a good choice for inferring sequential predictions based on previous states [180]. To tackle memory issues of processing large point clouds, small sections of the input point cloud are fed into the network one at a time, instead of processing all of it at once. In each small section, points in the queue are processed consecutively by REIN in two steps. 1) *Edge Prediction:* REIN tries to predict connections, i.e., edges, between the new vertex (which was chosen from the queue) and the current partially predicted mesh. Two RNNs are used for edge prediction: State RNN and Edge RNN. State RNN encodes the current state of the graph with its nodes and edges, given a point cloud and its latent vector as input. Edge RNN attempts to predict the sequence of edges considering the current state. 2) *Face Generation:* All of the vertices and predicted edges are investigated to form faces. However, the face generation module encounters problems generating surfaces from edge predictions, especially for nonmanifold surfaces. Qualitatively and quantitatively, REIN produces better mesh surfaces than BPA [10] and PSR [8].

D. Implicit Neural Representation

Neural networks are universal function approximators [181]; hence, they can be used to approximate any measurable function, including SDF and occupancy/indicator function, or to model other properties, such as radiance fields. Neural networks that parameterize such implicitly defined functions, without explicitly parameterizing the surface or properties of interest, are considered implicit neural representations [51].

Similar to implicit functions stored in discretized voxel grids, different functions can provide geometric information for parameterizing a surface by a neural network [123]. There are also other functions that focus on capturing surface-related properties, such as appearance, texture, or reflectance properties. In particular, these functions can be as follows.

1) Level set methods define a DF f on the entire point set and then extract the zero-level set $f = 0$ as the boundary of an input object, as illustrated in Fig. 18. They divide a 3-D space threefold into an interior part, an exterior part, and an exact overlap with the object's surface. Given a point (x, y, z), the function f calculates the distance of this point to the boundary of the object, specifies its sign (SDF) [182], and decides the location of the point w.r.t. the surface. The sign indicates whether a point is inside or outside of the surface. Therefore, in contrast to SDFs stored in voxels that discretize 3-D space and store SDF value in each voxel, SDFs in implicit neural representation are calculated for each point individually using a neural network. DeepSDF [50], which will be explained further down in this survey, was the first paper to propose this approach.
2) Occupancy functions model an approximate likelihood of whether a point is occupied by part of an object or not. This can be expressed as a binary classification problem to classify a point as occupied or unoccupied. The approach can be interpreted as a special case of SDF that only considers the sign of SDF values [50]. Occupancy networks [45] and

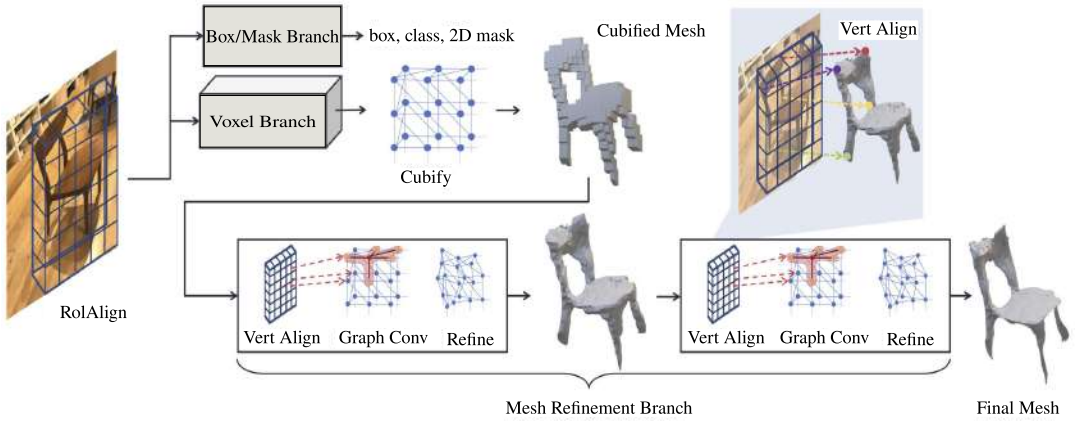


Fig. 17. *Mesh R-CNN [44] architecture. After the object detection step, the voxel branch predicts a coarse voxel representation for each object detected by Mask R-CNN [178]. Then, in the mesh refinement branch, the cubified object is transformed into the mesh after a series of refinement steps.*

IM-NET [49] fall into this category and will be clarified subsequently.

3) Radiance fields refer to a set of techniques that aim to model the radiance or appearance properties of an object or scene. Notable examples of these methods include NeRF [57] and its variants, and Sections VI-D2a and VI-D2b will provide thorough explanations of them.

1) Implicit Neural Representation Based on Variants of SDF or Occupancy Function: The key idea behind these implicit neural representations is to represent a shape as a neural network that takes a point in space as input and outputs some property of that space, i.e., mapping it to occupancy or signed distance of the shape at that coordinate. However, implicit neural representations cannot directly derive detailed 3-D shape features. Thus, an extraction step is needed to infer a corresponding explicit representation, such as a mesh. A possible isosurface extraction approach is the classical MC algorithm [125].

Compared to voxel-based representations, the memory cost of implicit neural representations remains constant with respect to the resolution. However, the capability of these methods to reconstruct fine details is constrained by the capacity of their underlying network architectures [51].

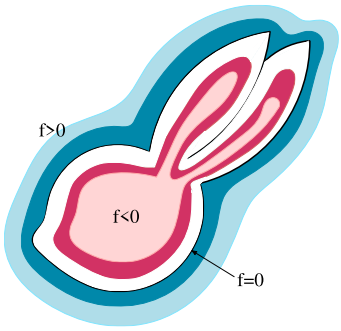


Fig. 18. *Level set methods divide a 3-D space into three parts: an interior part ($f < 0$), an exterior part ($f > 0$), and an exact overlap with the object's surface ($f = 0$).*

As mentioned previously, occupancy networks, IM-NET, and DeepSDF [45], [49], [50] represent pioneer works in implicit neural representation concurrently. Mescheder et al. [45] introduced a new representation for 3-D geometry, called occupancy networks, which can predict the continuous occupancy function using a neural network for the extraction of 3-D meshes. As illustrated in Fig. 19, the occupancy function is approximated with a DNN that determines an occupancy probability value between 0 and 1 for every possible point in 3-D point space (similar to a neural network for binary classification). The mesh is then generated from the occupancy network by utilizing a simple multiresolution isosurface extraction (MISE) algorithm, which employs octree structures and the MC algorithm [125]. This expressive approach does not require the discretization of 3-D space. The representation can be inferred from different kinds of input, such as single images, noisy point clouds, and coarse discrete voxel grids, and can encode various structures efficiently. In comparison to methods using different 3-D representations, such as 3D-R2N2 [20] (a voxel-based method), point set generating networks [30] (a point-based method), and Pixel2Mesh [38] and AtlasNet [34] as mesh-based techniques, occupancy networks show competitive qualitative and quantitative results for various inputs, e.g., single images, noisy point clouds, and coarse discrete voxel grids.

In a similar fashion, Chen and Zhang [49] attempted to solve 3-D shape analysis and synthesis problems by proposing an implicit field decoder (IM-NET), which is based on the application of binary classifiers. Based on two inputs, a point coordinate and a feature vector encoding a shape (extracted from a shape encoder), IM-NET specifies whether the point is inside or outside the surface, using only the sign of its SDF. They utilized their proposed implicit decoder as the decoder part of some conventional frameworks (such as autoencoders (AEs) and GANs) and

proposed IM-AE and IM-GAN, respectively. IM-AE and IM-GAN can be used for both 3-D reconstruction and shape generation tasks. Based on visual results, IM-AE generates smoother and high-quality surfaces compared to a classical 3-D-CNN-based decoder implementation, operating on voxelized shapes. IM-GAN showed better performance compared to AtlasNet [34] (in which output quality is constrained by the number of generated points) and 3D-GAN [23] (low coverage). For the single-view 3-D reconstruction task, the proposed framework constructs higher quality results than AtlasNet [34] and HSP [28]. However, applying the implicit decoder on each point in the training set increases training time considerably. In addition, the network does not generalize well to other categories since it is trained individually for each shape category.

With DeepSDF [50], a novel shape representation based on the concept of SDFs was introduced. Instead of storing SDF in a discretized regular grid, as done in classical surface reconstruction techniques, the network directly learns continuous 3-D models of SDF from point samples. The trained network predicts the corresponding SDF value of the input data, from which the zero-level set surface can be extracted. The zero isosurfaces can be rendered and visualized through raycasting or polygonization algorithms, e.g., MC [125]. The network takes (x, y, z) coordinates and a shape encoding vector as input to model a dataset of shapes. In order to obtain a meaningful latent space of shapes, an autodecoder is used for learning a shape embedding without an encoder. One of the advantages of the method is that the network size is considerably smaller compared to the voxel-based methods. DeepSDF outperforms Atlasnet [34] (a mesh-based method) and OGN [29] (an octree-based method) in reconstructing complex topologies with fine details. It further outperforms 3D-EPN [24] (SDFs stored in voxels) for the shape completion task.

Sitzmann et al. [51] introduced a novel architecture, called sinusoidal representation networks (SIRENs), an fc neural network that uses periodic sine as its nonlinearity for implicit neural representations. The motivation behind this lies in the fact that many recently published studies on implicit neural representation employing rectified linear unit (ReLU)-based MLPs are incapable of capturing high-frequency details of the input signal. There are two possible explanations for this phenomenon. 1) Conventional neural network architectures encounter difficulties while learning to apply the same function at two different coordinates, and thus, the learned functions are not shift-invariant in general. 2) ReLU nonlinearities cannot parameterize any signal that has information in its second derivative since its second derivative will be zero everywhere. Therefore, the authors suggested replacing conventional nonlinearities, such as tanh or ReLU, with a periodic sine activation function to improve final results. This replacement results in gaining a certain degree of shift-invariance and also addresses the problem of the second derivative since the derivative of sine is a shifted sine itself. The method was applied to a wide variety of areas, including image, audio, and video representations, 3-D reconstruction, and solving first- and second-order differential equations. In the 3-D shape reconstruction task, SIREN generates details of complex objects and scenes better than ReLU-based implicit representations, such as NeRF [57].

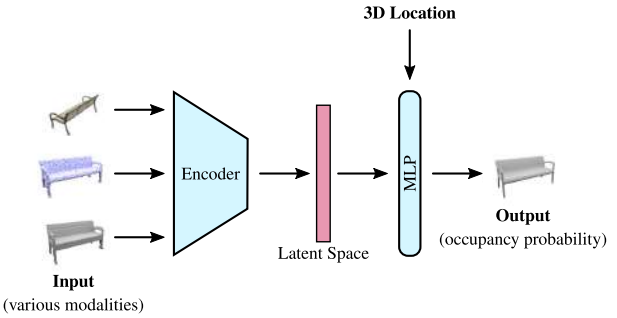


Fig. 19. *Occupancy networks architecture [45] predicts occupancy function for each point in 3-D space using a DNN. Different encoder architectures are used in occupancy networks depending on the task and input. A ResNet-18 architecture [173] for image input, a PointNet encoder [148] for point cloud input, and a 3-D-CNN for voxel input are employed.*

a) Methods based on unsigned distances: Some studies exploit unsigned distances instead of occupancy or signed distances for learning representations. With sign agnostic learning (SAL), Atzmon and Lipman [52] proposed a DL approach based on raw input data without any oriented normals or signs. Generally, regression-based methods utilize regression loss for training and need inside/outside ground-truth information for this process, such as DeepSDF [50] or occupancy networks [45]. In contrast to these methods, SAL uses a sign agnostic loss function that can be directly applied to raw unsigned data. The algorithm generates high-quality surfaces in comparison to AtlasNet [34] and a baseline method that approximates SDF based on the work by [9]. The D-Faust dataset, which comprises raw scans of humans in various poses, is used for the experiments. Although there is no need to include the signed implicit ground-truth representation in the calculation of the loss function during training and also closing surfaces for training data is unnecessary in this work, SAL predicts SDF as the final output, which also results in closing the gaps even in open surfaces and generating only closed outputs (closed surfaces, in this case, are a division of 3-D space into three regions: inside, outside, and on the surface of an object, and they do not have separate parts). Neural distance field (NDF) [53] is a method to predict the unsigned distance field for 3-D surfaces using a neural network. Similar to SAL, NDF does not close shapes during training. However, it can successfully generate open surfaces, shapes with inner structures, and open manifolds compared to IF-Net [56] and SAL [52].

DUDE [54] is another approach, which is able to represent a surface by combining the unsigned distance field

with the normal vector field. Evaluation of this method in comparison to DeepSDF and SAL demonstrates its superiority in producing high-quality outputs, especially for open surfaces, with visually pleasant renderings. The main difference between NDF and DUDE compared to SAL is that the first two can reconstruct both open and closed shapes with complex and detailed topology, while the latter attempts to close parts that should be open.

b) Part-based approaches: Encoding an entire surface into a single latent vector can lead to substantial information loss since the limited size and capacity of the latent representation causes accuracy and generalization issues [48]. In order to solve the difficulties of generalizing to other shape categories and scaling to large scenes, researchers resort to conditioning an implicit neural representation on local geometric features [46], [47], [48], [55], [56], [183], [184]. There are different approaches to the implementation of such conditioning. Some approaches fuse the volumetric representation (voxel grids) with the implicit neural representation and use local features stored in voxels for inferring implicit neural representation [46], [47], [55], [56]. Others use local patches to learn implicit neural functions [48], [183], [184]. All of these methods leveraged the advantages of encapsulating local and global information for proposing more generalizable and scalable approaches.

Jiang et al. [55] suggested the local implicit grid (LIG) representation, which decomposes 3-D space into a regular grid of overlapping part-sized local regions and encodes each region with implicit feature vectors. The key idea behind the algorithm is that objects in different categories share similar geometric features and details at neither microscale, i.e., a very small patch, nor macroscale, i.e., the entire object, but part scale. Therefore, a part-autoencoder was used to learn embeddings for different parts of an object and extract meaningful abstraction of its shape. The autoencoder consists of a 3-D-CNN encoder and an implicit network decoder in the form of a reduced version of the IM-NET [49] decoder. During inference, a pretrained implicit function decoder is used in each grid cell in order to generate the respective scene part. Eventually, the overlapping latent grids were optimized via the proposed mechanism to reconstruct the entire scene. Since this method generalizes shape priors learned from object datasets, it does not need any training on the scene-level dataset for reconstructing scenes from sparse oriented point samples. Therefore, it generates higher quality outputs from unseen object categories than other methods, such as IM-NET [49], since IM-NET learns only a single embedding for an entire object. Compared to traditional surface reconstruction methods such as PSR [8], [9], LIG is capable of recovering thin structures and details very well.

Likewise, Chibane et al. [56] introduced implicit feature networks (IF-Nets) that are composed of an encoding and a decoding tandem. The network takes voxels or point clouds as the input and predicts whether point p lies inside or outside of an object, resulting in a continuous surface at arbitrary resolution. To encode local and global structures of a 3-D shape, a 3-D multiscale grid of deep features is extracted instead of using a single vector to summarize an entire object. Consequently, rather than classifying (x, y, z) point coordinates directly, the decoder classifies a point based on these extracted features and creates occupancy predictions. IF-NET achieves better quantitative results than occupancy networks [45], point set generation network [30], deep MC [42], and IM-NET [49] in point cloud completion, voxel super-resolution, and single-view human reconstruction tasks. Moreover, Chibane and Pons-Moll [185] proposed an extension of IF-Nets for 3-D texture completion.

Peng et al. [46] developed convolutional occupancy networks, a hybrid voxel grid/implicit neural representation-based approach that combines convolution operations with implicit representations in the form of a convolutional encoder with an implicit occupancy decoder. The method is independent of the input representation. Given a point cloud or voxel grid as input, the method uses a 2-D plane encoder/3-D volume encoder based on PointNet to process the input by converting it into features and projecting these local features onto a plane(s)/volume. A convolutional 2-D plane decoder/3-D volume decoder further processes the feature plane(s)/volume using 2-D/3-D U-Nets [186], [187], integrating both local and global information. In the end, a small fc occupancy networks [45] is used to predict the occupancy probability from a given query point p and its feature in 3-D space. For rendering and extracting meshes from the input, the MISE algorithm is applied during inference. Evaluation of both object- and scene-level reconstructions was performed using synthetic and real-world datasets. The major difference between the novel method [46] and the original occupancy networks [45] is that convolutional occupancy networks capture the local features of the space and global features, leading to higher generalizability, scalability, and faster training. Moreover, it benefits from the translational equivariance property of convolutional networks while not supporting the rotational equivariance property.

In a similar work, Chabra et al. [47] introduced deep local shapes (DeepLSs), a method for deep shape representation, which uses learned local shape priors. As illustrated in Fig. 20, the key idea is the decomposition of a shape into small components in order to improve reconstruction results. To this end, local information of these components is stored in a grid of independent latent codes. Based on these, SDFs are predicted by applying DeepSDF [50] as a local shape neural network to each grid cell. DeepLS outperforms DeepSDF in accuracy and inference time by approximately an order of magnitude.

Unlike occupancy networks [45] and DeepSDF [50], which extract the global latent code vector from the entire input, local patches are modeled as deep implicit functions in patch-based approaches [48], [183], [184]. Erler et al. [48] presented a patch-based learning framework, called Points2Surf, which generates accurate implicit

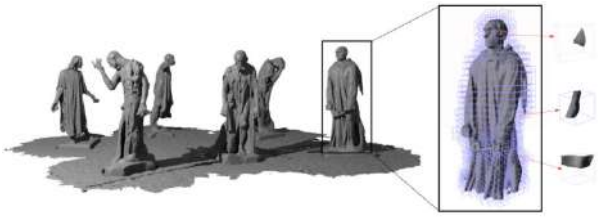


Fig. 20. *DeepLS [47] decomposes a scene into local shapes and uses a set of locally learned continuous SDFs defined by a neural network.*

surfaces directly from raw point clouds without surface normals. The underlying algorithm is based on the notion of considering a shape as a collection of small shape patches. Instead of representing an entire surface as a single latent vector, Points2Surf creates separate feature vectors for different patches to describe local details in addition to global information. By decomposing the surface reconstruction problem into learning a global function (that learns the sign of SDF) and a local function (that learns the absolute distance field of SDF with respect to local patches), Points2Surf succeeds in being robust to noise and missing parts and also generalizing well to unseen shapes. In addition, Points2Surf yields a significant drop in the reconstruction error on unseen classes compared to both data- and nondata-driven methods, such as DeepSDF [50] and AtlasNet [34], or SPR [9]. However, this patch-based approach results in longer computation time, inconsistencies between outputs of neighboring patches, and nonwatertight and bumpy surfaces.

There are a growing number of studies based on implicit neural representation for various tasks. Some authors investigated 3-D human reconstruction [184], [188], [189], 4-D reconstruction [190], and 3-D reconstruction of the appearance and texture of surfaces in addition to their geometry with 3-D supervision [191], [192] or without 3-D supervision [57], [193], [194], [195]. These are some recent articles [196], [197], [198], [199], [200], [201], which are SDF-based, and some [184], [188], [189], [202] that are based on predicting occupancy probability.

c) Equivariant neural networks: Chatzipantazis et al. [203] introduced an SE(3)-equivariant coordinate-based attention network called TF-ONet for 3-D surface reconstruction. Local shape modeling and equivariance are the two core design principles of this method. SE(3) stands for special Euclidean group in three dimensions representing transformations including translations and rotations in 3-D. In simple terms, equivariance means that, when the pattern in the input changes, i.e., when it is rotated or shifted to a specific direction, the output should also change in an equivalent proportion. TF-ONet works directly on unoriented and irregular point clouds and outputs the occupancy field of a shape. To predict the occupancy score at any given point in space, TF-ONet creates equivariant features for each point that function as keys and values of specialized attention blocks. This enables TF-ONet to output high-quality reconstructions and generalize to novel scenes composed of multiple objects, despite being trained on single objects in canonical poses. Inspired by SE(3) transformers [204] and tensor field networks [205], TF-ONet attention modules ensure equivariance by incorporating symmetries into the learning process. It is basically a two-level approach. 1) The first level, i.e., an encoder, applies self-attention in local neighborhoods around each point to infer local features from the point cloud. 2) The second level, i.e., a cross-attention occupancy network, uses the extracted point features and the coordinates of a query point in space to calculate the value of the occupancy function for the specific query point.

For single-object reconstruction tasks, TF-Onet performs comparably better than nonequivariant networks, such as occupancy networks [45], convolutional occupancy networks [46], IF-Net [56], and also equivariant networks, such as vector neurons [206] and GraphOnet [207] considering evaluation metrics, such as Chamfer-L1, F1-score, and IoU. For scene reconstruction tasks trained only on single objects, global shape modeling-based techniques, such as occupancy networks [45] and vector neurons [206], are not able to generalize to scenes containing multiple objects. Moreover, local shape modeling-based methods, such as convolutional occupancy networks [46], which are not equivariant under SE(3) transforms, are only able to produce low-quality objects in novel poses. TF-ONet instead excels at the tasks and can generalize to novel scenes with high quality, benefiting both from local shape modeling and equivariant properties.

2) NeRF-Based Approaches:

a) Fundamentals of NeRF: NeRFs [57], commonly referred to as NeRFs, are basically used for view synthesis. The main idea behind NeRFs is to train a model that can produce new views of a scene or an object and can represent them in 3-D, given a set of 2-D images from different viewing angles as input. Hence, multiple input views of a scene and their corresponding camera poses are used to render new views of that scene by interpolating between the given views. The NeRF method employs an fc deep network to represent a scene. Each input (x, y, z, θ, ϕ) is a single continuous 5-D coordinate that encompasses spatial position and viewing direction, and each output $(\text{RGB}\sigma)$ is density and view-dependent emitted radiance at that particular spatial location. Consequently, the neural network describes an implicit function that exists throughout all locations as a continuous representation without any discretization. As a result, by implicitly encoding density and color through a neural network, NeRF has demonstrated impressive performance on new view synthesis of a particular scene.

Although overfitting is usually an undesirable behavior in machine learning, the key part of this approach is the usage of a neural network that is overfitted to one

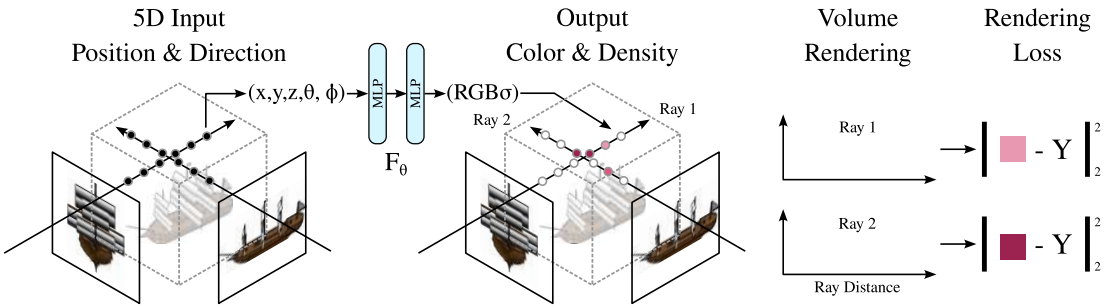


Fig. 21. *Overview of NeRFs [57]. The ship in the figure is borrowed from the ShapeNet dataset [96].*

particular scene and only cares about this specific scene. For rendering a new scene, it is necessary to take a fresh neural network and train it from scratch until it is overfitted to the new scene. Therefore, instead of storing a scene as a mesh or a voxel grid, the scene is stored in the weights of the neural network. For instance, if a scene consists of a tree, the weights represent this tree and are very specific to it, outputting nonsense for another scene if not being trained once again.

To explain the fundamentals of NeRF in more detail according to Fig. 21, the images have to be transformed to 5-D coordinates (x, y, z, θ, ϕ)s first. (x, y, z) are coordinates of a pixel point in 3-D space, and (θ, ϕ) are related to the viewing angle. For each pixel on an image, a ray is sent through. Therefore, every pixel in every input picture defines a ray, and then, it is sampled along the ray. Consequently, each input image sends out a lot of rays, and for each ray, there are many sampled points. Next, for each location represented as (x, y, z, θ, ϕ), the neural network effectively determines the presence of an object and subsequently identifies its corresponding color. This nine-layer fc network provides four numbers (RGBσ) as the output: the (RGB) is the color of that particular pixel point, and σ is the density for each of the individual points. The density value serves as an indicator of the presence or absence of an object in the designated region of space, as well as its density. If this process is done for all the points in space from all viewing angles, a complete 3-D representation of what it looks like can be inferred. The neural network outputs different results for the same location depending on different viewing angles. Accordingly, it can capture the reflections, lighting effects, and transparency. Eventually, classical techniques for volume rendering are employed to project the network outputs onto a 2-D image. Given that volume rendering is intrinsically differentiable, it is possible to define a loss function that measures the difference between the predicted and the ground-truth color of the ray. In order to convert NeRF to a mesh, MC can be further applied.

To produce high-resolution complex scenes, two interesting tricks were utilized: 1) positional encoding; 2) hierarchical sampling. Positional encoding, which is similar to the same one in transformers [208], is used to map the 5-D input vector to higher dimensional space using sin and cos waves, helping MLP in approximating and representing high-frequency functions. It enhances the ability of a neural network to not only capture coarse-grained structures but also to perform well in representing finer details. Hierarchical sampling is a two-step sampling method with two networks: a coarse network and a fine network. The points on the ray are sampled in a uniformly distant fashion from each other. These sampled points are run through the network for density prediction. Next, an evaluation step is taken place to decide where should be sampled more in the second round, based on the output of the previous step. Thus, the output of the coarse network discloses where the important stuff is. The second round of sampling starts with points with higher density, i.e., points closer to the particular object that is perceived, and the vicinity of such points will be sampled a lot more. Both coarse- and fine-grained networks are optimized at the same time using a loss.

Delving into the advantages associated with NeRFs, it is clear that these methods are not view-dependent, without the need for any 3-D input supervision. In addition, NeRFs are memory-efficient compared to voxel grid representation. One neural network of one scene fits into a few megabytes, which might even be smaller than the input image size for that scene, whereas dozens of gigabytes might be needed for storing the same scene in voxels. Regarding the limitations of NeRFs, what makes them impractical is their requirement for a large number of high-quality posed images as input. The more images are fed, the better the output quality will be. Another downside is related to their high computational cost, originating from optimizing each scene individually without sharing knowledge between different scenes [62]. This implies that, for every scene, the network should be trained again, and a pretrained one cannot be utilized. For instance, it takes around 100k–300k iterations, i.e., roughly one to two days, for the naive NeRF network [57] to be trained on a single scene using a single NVIDIA V100 GPU.

b) NeRF and its variants for view synthesis: This section provides a summary of some of the papers that aim to enhance NeRF and its abilities. In NeRF++, Zhang et al. [61] analyzed NeRF and uncovered three major problems and situations in which NeRF might fail: shape-radiance ambiguity, near-field ambiguity, and

parameterization of unbounded scenes, such as large real-world scenes. The first two issues are related to the fact that NeRF is actually overparameterized, i.e., the degree of freedom for NeRF to hallucinate and move toward a completely wrong answer is high. However, the authors of NeRF [57] use an interesting implementation trick and regularization. They feed viewing angles in the very last layers of the MLP network. Therefore, the MLP actually starts with locationwise coordinates of a point in the beginning, and viewing angles are fed in the last layers, resulting in a limited degree of freedom for NeRF. Accordingly, if all 5-D coordinates are fed to the network from the beginning, the shape radiance ambiguity becomes a big issue, affecting the quality of NeRF's outputs drastically.

NeRF++ proposes a couple of solutions to tackle these three problems and enhance output quality. By introducing an auxiliary loss, NeRF can avoid moving toward a poor solution, which may lead to completely wrong scene geometry estimation, thus addressing the shape-radiance ambiguity issue. Furthermore, adaptive near-field culling is proposed to solve the near-field ambiguity issue. It culls the front part of each view frustum adaptively based on the geometry of a scene, i.e., it prevents estimating the geometry right in front of the camera contrary to vanilla NeRF. The third issue concerns scenarios in real-world settings where precise reconstruction of objects in front of the camera is essential. However, the camera's ability to capture other items beyond these objects necessitates a certain level of reconstruction for the distant items as well. NeRF++ suggests homogenous parameterization that enables having a detailed reconstruction in the foreground and a detailed reconstruction of the background. This is done by training two NeRFs, one for the foreground part of the scene and the other for the background part, increasing the capacity of the model for reconstructing details at different levels. NeRF++ still needs per-scene training, and one scene takes about three days to be trained.

PixelNeRF [62] is built upon the concept of NeRFs for 3-D reconstruction and synthesizing photorealistic 3-D scenes from a single or a small number of posed images. PixelNeRF attempts to tackle the requirement of NeRFs for a lot of images as the input and make it generalizable. Considering the fact that extracting 3-D geometry and the appearance of a scene from limited input is a challenging task, and NeRFs do not share knowledge between the scenes, the framework proposes to condition an NeRF on spatial image features. Thus, pixelNeRF employs a fully convolutional image encoder that infers a pixel-aligned feature grid. Then, a spatial location and its corresponding encoded feature are fed to an NeRF network for color and density prediction. PixelNeRF shows better generalization capabilities and performance compared to NeRF. However, its rendering time is still slow, and more input views cause a linear increase in the runtime.

In another concurrent work to overcome the generalizability issue and long optimization time of NeRFs, MVSNeRF [63] suggests a DNN that can reconstruct an NeRF, given only three nearby input views. This approach combines plane-swept cost volumes, which are used for geometry-aware scene reasoning in MVS, with NeRF models. To create a cost volume, MVSNeRF first warps 2-D image features onto a plane sweep. Then, a 3-D-CNN is leveraged for the reconstruction of a neural encoding volume with per-voxel neural features. Next, features interpolated from the encoding volume are employed to predict density and RGB radiance for an arbitrary point using an MLP. Achieving comparable or better rendering results, MVSNeRF can significantly surpass NeRFs [57] in terms of optimization time efficiency, i.e., roughly 30 times faster, if more images are provided as input. Moreover, it generalizes better than PixelNeRF [62] and IBRNet [209].

MipNeRF [210] attempts to address one of the problems of NeRF, which is the production of blurred or aliased renderings when dealing with training or testing images at different scales. In NeRF, all of the cameras have the same distance from an object. Thus, it is able to do view synthesis without the need to think about scaling or aliasing. However, when new cameras are to be added at different scales, NeRF begins to collapse since it is a single-scale model trying to tackle a multiscale problem. To fix this issue, MipNeRF proposes some modifications to the vanilla NeRF including the following: 1) casting a cone instead of sending a ray through each pixel; 2) slicing up the cone into conical frustums instead of sampling single points along each ray; 3) computing integrated positional encoding instead of positional encoding of a single coordinate along the ray; 4) in general, training a single neural network that describes the scene at multiple scales instead of training separate neural networks at various scales. These new properties help MipNeRF reason about the scale of its inputs. MipNeRF is capable of producing high-resolution renderings across multiple scales rather than just at a single scale in vanilla NeRF. NeRF's performance decreases when being trained on multiscale data, while MipNeRF's performance does not. The number of parameters in MiPNeRF is half of that in NeRF while also being 7% faster for their multiscale dataset. Mip-NeRF360 [211] and ZipNeRF [212] are some other recent methods used for antialiasing NeRFs.

In a work proposed by NVIDIA, instant NGP, Müller et al. [213] try to facilitate and speed up neural graphics primitive tasks. A neural graphics primitive is an object represented by a neural network that takes a query as input, such as position and some extra parameters, and outputs appearance and shape attributes. Examples of NGP can be computing SDF, NeRFs, radiance cashing, and so on. To bring about simplicity, instant training, real-time rendering, and high-quality results for instant NGP, solutions such as multiresolution hash encoding by storing the trainable feature vectors in a compact spatial hash table, using a small neural network called a fully fused neural network, and improvement of

Table 2 Comparison of Various 3-D Reconstruction Methods: CNN, GCN, IoU, CD, EMD, HD, Binary Cross-Entropy (BCE), AP, Cross-Entropy (CE), Squared Distance Error (SDE), NC, Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM)

Name	Input	Output	Method	Loss	Dataset	Metric
3D-R2N2 [20]	A single-view image or multi-view images	Voxels	Encoder-decoder, 3D conv LSTM	Voxel-wise CE	ShapeNet, Pascal 3D [229], Online products [230]	IoU, CE
VRN [21]	Voxels	Voxels	Encoder-decoder	BCE	ModelNet	Accuracy
TL-embedding [22]	RGB images	Voxels	Encoder-decoder	CE, Euclidean loss	ShapeNet, IKEA dataset [231]	AP
3D-GAN [23]	An image	Voxels	CNN, 3D GAN, Encoder-decoder	BCE loss, KL-divergence loss, reconstruction loss	ModelNet, IKEA dataset, ShapeNet	AP
3D-EPN [24]	Depth maps	Voxels	Encoder-predictor network	L1 loss	ShapeNet	Accuracy, L1 error
3D shape completion [25]	Point cloud and a 3D bounding box	Occupancy grid or SDFs	Encoder-decoder	Reconstruction loss, maximum likelihood loss	ShapeNet, KITTI, ModelNet	Hamming distance, accuracy, completeness
SG-NN [139]	RGB-D scan	A sparse TSDF	Encoder-decoder	L1 loss, BCE loss	Matterport3D	L1 error
Octnetfusion [26]	One/multiple 2.5D depth image(s)	Voxel or Octree	Encoder-decoder	L1 loss, BCE	ModelNet40	IoU, precision, recall
HSP [28]	RGB images or depth images or partial grids	Voxels	Encoder-decoder	CE	ShapeNet	IoU, CD
OGN [29]	Voxels	Structure of an octree and binary occupancy map	Encoder-decoder	CE	ShapeNet	IoU
Adaptive O-CNN [27]	A single image or point cloud	A patch-guided adaptive octree	Encoder-decoder	CE, SDE	ModelNet, ShapeNet	CD, accuracy
Point set generation net [30]	A single RGB or RGB-D image	Point cloud	Encoder-predictor	CD, EMD	ShapeNet	IoU, CD, EMD
Latent 3D points [31]	Point cloud	Point cloud	Encoder-decoder, GAN	CD, EMD	ShapeNet, ModelNet	JSD, coverage, MMD
FoldingNet [32]	Point cloud	Point cloud	Encoder(graph-based)-decoder	CD	ShapeNet, ModelNet	Accuracy
PointFlow [33]	Point cloud	Point cloud	Encoder-decoder	Prior loss, reconstruction loss, posterior loss	ShapeNet	JSD, MMD, coverage, CD, EMD, accuracy
AtlasNet [34]	2D images or point cloud	Mesh	Encoder-decoder	CD loss	ShapeNet	CD
Meshlet [37]	Point cloud	Mesh	Encoder-decoder	CD loss	ShapeNet	CD, HD
Pixel2Mesh [38]	An RGB image	Mesh	Graph convolution network	CD loss, normal loss	Dataset of 3D-R2N2 [20]	F1-score, CD, EMD
Pixel2Mesh++ [39]	A few RGB images or multi-view images	Mesh	GCN	CD loss, normal loss	Dataset of 3D-R2N2 [20]	F1-score, CD
CMR [40]	An image	Mesh	Convolutional encoder	Reprojection loss, regression loss	CUB-200-2011 dataset, PASCAL 3D+ dataset	IoU
Point2Mesh [41]	Point cloud	Mesh	CNN	CD loss, beam-gap loss	A large dataset of object scans [232]	F1-score
DMC [42]	Point cloud	Mesh	Encoder-decoder network with skip connections	Point to mesh loss, occupancy loss, smoothness loss, curvature loss	ShapeNet	CD, accuracy, completeness
Scan2Mesh [43]	One/multiple depth image(s)	Mesh	CNN, graph neural network	CE, CD loss	ShapeNet	CD, normal deviation

Table 2 *(Continued.)* Comparison of Various 3-D Reconstruction Methods: CNN, GCN, IoU, CD, EMD, HD, Binary Cross-Entropy (BCE), AP, Cross-Entropy (CE), Squared Distance Error (SDE), NC, Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM)

Mesh R-CNN [44]	An RGB image	A category label, segmentation mask, boundary box, a 3D triangular mesh	GCN	BCE, CD	ShapeNet, Pix3D dataset	F1-score, CD, NC
Meshing point clouds with IER [35]	Point cloud	Mesh	CNN	Euclidean distance, geodesic distance	ShapeNet	F1-score, CD, NC
REIN [36]	Point cloud	Mesh	Encoder-decoder, RNN	CD, BCE	ShapeNet, ModelNet10	CD, point normal similarity
Occupancy Nets [45]	An image or point cloud or discrete voxel grids	Implicit surface	Encoder, fully connected network	CE	ShapeNet, KITTI, Pix3D	IoU, CD, NC
IM-Net [49]	Images or voxels	Implicit surface	Encoder-decoder, GAN	Weighted mean squared error, Wasserstein GAN loss	ShapeNet	MSE, IoU, CD, LFD, MMD, COV
DeepSDF [50]	Point cloud	Implicit surface	Auto-decoder	L1 loss	ShapeNet	CD, EMD, accuracy
SIREN [51]	Point cloud	Implicit surface	Fully connected neural network	SDF loss (Eikonal equation)	Stanford 3D scanning repository	N/A
SAL [52]	Point cloud or triangle soups	Implicit surface	Variational encoder-decoder	Sign-agnostic loss with L2 distance	D-Faust dataset	CD
NDF [53]	Point cloud	Implicit surface	Encoder-decoder	Unsigned distance field loss	ShapeNet	CD
DUDE [54]	Triangle soups	Implicit surface	Feed-forward networks	L2 loss	ShapeNet	IoU, mean absolute error, normal map error
LIG [55]	Point cloud	Implicit surface	Encoder-decoder	BCE loss	ShapeNet, Matterport 3D, SceneNet	F1-Score, CD
IF-NET [56]	Point cloud or occupancy grid	Implicit surface	Encoder-decoder	CE	ShapeNet	IoU, CD, NC
Conv occupancy nets [46]	Point cloud or voxels/coarse occupancy grid	Implicit surface	Encoder-decoder	BCE	ShapeNet, ScanNet, Matterport 3D	F1-score, IoU, CD, NC
DeepLS [47]	Depth data or mesh	Implicit surface	Autodecoder network	Negative log likelihood loss	Stanford 3D scanning repository, 3D warehouse [233]	CD
Point2surf [48]	Point cloud	Implicit surface	Encoder-decoder	L2 loss, BCE loss	ABC dataset	CD
UNISURF [223]	RGB images	Implicit surface	MLP	Reconstruction loss, Surface regularization	DTU [234], BlendedMVS [235], SceneNet	CD
NeuS [222]	RGB images	Implicit surface	MLP	Color loss, regularization loss, mask loss	DTU [234], BlendedMVS [237]	CD, PSNR, SSIM

training and rendering algorithm are proposed as main ideas.

The amount of research efforts based on NeRF is increasing. From relighting [64], [65], [214], [215] and view synthesis without pose supervision [216] to learning nonrigid objects and dynamic scenes [66], [67], [68], [217], [218], [219] and tackling computational challenges of NeRF and heading toward the real-time rendering [58], [59], [60], [220], numerous studies have been conducted to broaden the horizons of NeRF and its various applications.

c) NeRF for 3-D surface reconstruction: In an NeRF model, the scene geometry is hidden inside the neural networks, i.e., it is implicit. In order to achieve 3-D surface reconstruction and transform the NeRF representation into an explicit representation, such as a mesh, a surface extraction step is essential. By analyzing and thresholding the learned density, i.e., extracting an arbitrary level set of the density function that is learned by NeRF, and using methods such as MC, the baseline NeRF can extract and reconstruct an approximate explicit 3-D geometry [221].

Table 3 Quantitative Report About Some of the Methods' Performance on ShapeNet. CD, IoU, AP, and F_Score Are Calculated as the Average. For IoU, F_Score, and AP, the Higher the Better. For CD, the Lower the Better. The Number of ShapeNet Categories Used in an Experiment (#Cats), Not Measured or Not Mentioned (-), SVR, MVR, Reconstruction (R), Completion (C), Autoencoding (AE), Training Time (T), Inference Time (I), Generating a Mesh (mg), Memory (Mem.), and MS. * Is Calculated for (32^3). + Is Related to Chamfer-L1. For Detailed Information Regarding Data Preparation Methods, Train/Test Splits, Metrics, and Other Specific Details, Please Refer to the Context of Each Individual Paper

Papers	ShapeNet					Task	Time	Size
	CD	IoU	F_score	AP	#Cats			
3D-R2N2 [20]	-	>0.60	-	-	13	MVR (3 views)	-	-
TL-embedding [22]	-	-	-	65.40	5	SVR	-	-
OGN [29]	-	0.59 *	-	-	13	SVR	5 days T for 256^3	0.54G(256^3)
Adaptive O-CNN [27]	0.00460	-	-	-	13	SVR	-	-
Point set generation [30]	0.25000	0.64	-	-	13	SVR	-	-
PointFlow [33]	0.00070	-	-	-	13	AE	-	-
AtlasNet [34]	0.00150	-	-	-	13	AE (25 patches)	-	-
AtlasNet [34]	0.00510	-	-	-	13	SVR (25 patches)	-	-
Meshlet [37]	0.00900	-	-	-	-	R	-	-
Pixel2Mesh [38]	0.59100	-	59.72	-	13	SVR	72h T - 15.5ms I (mg)	-
Pixel2Mesh++ [39]	0.48000	-	66.48	-	13	MVR (3 views)	96h T - 15.5ms I (mg)	-
Scan2Mesh [43]	0.00160	-	-	-	8	C	-	-
Meshing with IER [35]	0.00071	-	87.20	-	8	R	<10s I/a pc with 12,800 pts	-
IM-Net [49]	0.00140	-	-	-	5	SVR	-	-
IM-Net [49]	0.00060	0.75	-	-	5	AE	-	-
DeepSDF [50]	0.00030	-	-	-	5	AE	9.72s I	0.0074(MS)
DeepSDF [50]	0.00160	-	-	-	3	C	9.72s I	0.0074(MS)
IF-NET [56]	0.00002	0.88	-	-	13	R	-	-
Occupancy Nets [45]	0.21500+	0.57	-	-	13	SVR	3s I/per mesh	-
Occupancy Nets [45]	0.07900+	0.77	-	-	13	C	3s I/per mesh	-
Conv onets [46]	0.04800+*	0.87	93.30	-	13	R	-	5.9G Mem.

Although NeRF and its variants generate impressive results for the novel view synthesis task, they cannot output high-quality 3-D surface reconstruction. The quality of the extracted 3-D geometry is not satisfactory because the initial objective of NeRF is novel view synthesis, not 3-D surface reconstruction. Since the density-based representation used in NeRFs is flexible and does not have enough constraints on 3-D geometry [222], it imposes some limitations on inferring accurate surface geometry, especially when ambiguities exist. Therefore, the extracted surfaces usually contain artifacts. To alleviate this issue, some papers have been presented for the 3-D surface reconstruction task that tried to incorporate implicit neural surface representation approaches based on an SDF or an occupancy function into NeRF-based methods, benefiting from the advantages of both categories. In these methods, instead of choosing the density-based scene representation used in NeRF, the scene space is usually represented as an SDF or an occupancy function.

Oechsle et al. [223] proposed UNIfied Neural Implicit SUrface and Radiance Fields (UNISURFs), which is a framework for 3-D surface reconstruction and capturing high-quality implicit surface geometry from multiview images without the need for object masks. It unifies the implicit surface models with radiance fields for solid and nontransparent object reconstruction given a set of RGB images. UNISURF represents surfaces and defines object or scene geometries using occupancy values. It learns and optimizes this implicit surface via a volume rendering method like NeRF. The output mesh is extracted using the MISE algorithm [45]. Considering reconstruction quality, UNISURF outperforms NeRF [57]. There are some limiting factors for this method, including reconstructing only solid objects and constraints to model transparencies, performance drop for overexposed or rarely visible regions in the ground-truth images, and inability to resolve the shape-appearance ambiguities, such as shadows and holes in objects.

In another concurrent attempt, Wang et al. [222] presented NeuS that learns neural implicit surface representation based on SDF using volume rendering, with the goal of reconstructing the 3-D surface of an object or scene given multiple images from different viewing points without leveraging mask supervision. Instead of just doing standard volume rendering or standard surface rendering, this framework suggests using volume rendering (inspired by NeRF) in addition to surface representation with neural SDF. The key idea behind this method is to represent a 3-D surface as the zero-level set of an SDF, i.e., representing a surface with neural implicit SDFs, and to introduce a new volume rendering method by taking inspiration from NeRF, for training a neural SDF representation with robustness. This novel volume rendering technique attempts to learn the weights of the neural network by rendering images from the implicit SDF first and then minimizing the difference between the rendered images and the input images. NeuS performs quantitatively and qualitatively better than NeRF [57] and UNISURF [223] in high-quality surface reconstruction. However, one failure case of NeuS is its inability to accurately reconstruct textureless regions. This limitation is caused by the ambiguity of these textureless regions for reconstruction in neural rendering.

Variants of NeuS [224], [225] have been proposed with the goal of improving the reconstruction quality. HF-NeuS [224], a method for multiview surface reconstruction with high-frequency details, breaks down the SDF

into fundamental components, namely, base and displacement functions, and adopts a gradual increase in high-frequency details through a coarse-to-fine strategy. In Geo-Neus [225], by utilizing sparse 3-D points in SfM constraint in conjunction with the photometric consistency in MVS constraint, the learning of neural SDF can be enhanced.

In a similar fashion to NeuS, another concurrent work called VolSDF [226] suggested a volume rendering framework for implicit neural surfaces. Replacing general-purpose MLP densities with densities from a certain family, i.e., in this case representing the density as a function of the signed distance to the scene's surface, is the core contribution of VolSDF. Two fc neural networks, one for the approximation of the SDF of the learned geometry and the other for representing the scene's radiance field, form the structure of this framework. Compared to NeRF [57] and NeRF++ [61], VolSDF generates more accurate results. One of the limitations of VolSDF is that it assumes the object is homogeneous with a constant density. Moreover, its reconstruction time is still high due to the independent training of the network for each scene.

Recently, SDFStudio [227], which is a framework for neural implicit surface reconstruction, has been released. It is built on top of nerfstudio [228] and includes a unified implementation of VolSDF, NeuS, and UNISURF, three popular neural implicit surface reconstruction techniques. Because of the unified and modular implementation of this framework, transferring ideas between methods is simple. The idea from Geo-NeuS can be integrated with VolSDF, bringing about the Geo-VolSDF method.

VII. DISCUSSION AND FUTURE TRENDS

In Section VI, the latest attempts toward 3-D reconstruction using DL techniques were reviewed. A summary and comparison of presented learning-based surface reconstruction approaches can be found in Table 2. Furthermore, Table 3 contains a quantitative report about the performance of some of the approaches on the ShapeNet dataset. There is a qualitative gap between 3-D models created by learning-based approaches and artist-created CAD models [43], and there are still open problems in this scope. Some of these challenges are listed in the following.

In the existing approaches, serious bottlenecks are caused by computation time and generalization power. The requirement of long training time is a drawback to the adoption of some of the DL-based approaches. On the other hand, there are concerns raised about the environmental impact of prolonged training periods. To this end, designing models with a reduced number of parameters, less complexity, and yet high performance can be a target to hit. In addition, the utilization of transfer learning may serve as a partial solution. Regarding the generalizability issue, methods with the capability of multicategory generalization, i.e., generalizing well to other topology categories, should be further investigated. One solution might be to learn latent shape spaces that are not class-specific. Consequently, as a future direction, moving toward models with comparable shorter training time and stronger generalizability can be an interesting yet reasonable strategy.

Current methods are highly dependent on an external supervisor for annotating input data. Reducing the need for supervision is a desirable trait for a learning-based approach [40]. Furthermore, there are various large-scale datasets appropriate for geometric DL tasks. However, there is still a need for creating datasets with richer 3-D annotations that are suitable for shape and surface reconstruction.

On the other hand, some of the current evaluation metrics fall short of capturing surface properties accurately. Therefore, it is necessary not to be limited to quantitative results but to explore qualitative results to gain a deeper understanding of surface details as well. Moreover, presenting better and more robust evaluation metrics, which are at the same time computationally efficient and less complex (in point cloud comparison, CD has quadratic complexity for instance), is another area that is essential to focus on.

In the context of volumetric methods, various challenges exist that should be tackled. Because of the discretization of data, some input information and details may partially be lost. Cubic growth in memory and computational costs with respect to resolution and poor scalability of these methods with resolution increase lead to difficulty in inferring high-resolution outputs. Considering the influence of 3-D resolution on the performance of volumetric CNNs for instance, better performance can be achieved by designing efficient volumetric CNN architectures for instance, which are able to scale to higher resolutions [128].

For point-based approaches, current methods extract a fixed and limited number of points from the point cloud dataset and feed them to their network architecture, thus affecting the output quality. Overcoming this limitation and implementing models with the ability to handle variable-length input can be ambitious yet interesting future directions.

In mesh-based approaches, it is challenging to define a loss on meshes, which is easy to optimize [34]. One of the limitations of patch-based approaches in the mesh-based representation category that affects the reconstruction of fine details is the usage of a fixed scale mesh patch [37]. A coarse-to-fine approach and extracting mesh patches at different scales might result in more precise outputs. On the other hand, generating a closed shape using patch-based methods, and recognizing and segmenting shapes using these methods are issues that still require solutions [34].

Implicit neural representations have recently gained popularity due to their performance and favorable properties. Existing isosurface extraction approaches used for extracting representations from implicit neural representations are computationally intensive and, thus, comprise a bottleneck. Furthermore, it may be worthwhile to combine sign-agnostic implicit neural approaches with

generative methods, such as GANs [52]. Moreover, NeRF-based approaches mostly suffer from high computational cost, long training time, and the inability to share knowledge between various scenes, thus being scene-specific networks. The necessity for more input images in order to have high-quality outputs should be alleviated. Improving NeRF-based methods' time and computation efficiency, their generalizability to unseen scenes, and their surface reconstruction ability can be important research questions.

In general, reducing the performance gap between synthetic and real-world data, proposing better and more representative evaluation metrics for quantifying shape reconstruction analysis results [49], conducting research in the challenging task of scene-level reconstruction, empowering proposed methods with multiscale reconstruction (coarse-to-fine manner) [48], implementing and employing methods for capturing high-frequency details with the purpose of reconstructing thin parts of a scene or object in high-quality, considering the equivariance concept for designing a neural network, and fusing different approaches mentioned in Fig. 6 in order to enjoy the benefits of them simultaneously are aspects that should not be ignored in future studies. In addition, the application of transformer architectures [208], i.e., a DL model that is based on the self-attention mechanism, seems to be promising in 3-D vision [237], [238], [239]. On the other hand, self-supervised learning [240], which is a technique for predicting unobserved or hidden part of the input from observed or not hidden part of the input, can be one of the interesting approaches for solving reconstruction and in general computer vision problems with low quality and a limited amount of data. Furthermore, considering the current interest, diffusion models [241], [242], [243], [244], which learn to infer and generate a meaningful output from pure noise, seem to be another exciting approach to be used in 3-D generation, completion, and reconstruction [245], [246].

It is equally expected that surface reconstruction applications play an increasingly important role. One of the major uses will be in observational RS-related disciplines where surface reconstruction will aid in archeological discoveries, agriculture, disaster prevention and response, and cartography. Equally, design- or projection-based applications have great utilization potential for learned surface reconstruction, including, but not limited to, 3-D modeling in games and movies, architecture, or CAD. Yet, all of the aforementioned scenarios are considering only (close to) static surfaces. The anticipation is that accurate reconstruction of dynamically changing objects and environments, nonrigid objects or scenes, textureless regions and transparent objects, and overcoming the challenges of rarely visible regions, occlusions, shadows, and holes in an object or scene will be crucial and consequential next steps in this field of study. Overall, more applications of neural learning approaches will emerge for surface reconstruction, especially in SFX and VFX animation, human reconstruction, robotics, autonomous driving, and medicine.

VIII. CONCLUSION

In this article, we provided a review of the state-of-the-art approaches for learning-based 3-D surface reconstruction. We have taken no special perspective, making the manuscript accessible not only to method researchers but also to applied users seeking to contextualize these approaches for their domains.

For this, we have reiterated commonly used open and accessible benchmarking datasets, different input and output data modalities, and some acquisition techniques. To make processing results comparable, we have highlighted widely used metrics for evaluating learned models and detailed their particularities.

The main part of this article has introduced DL-based 3-D surface reconstruction approaches. In summary, these can be classified into four major categories based on their output representations: 1) voxel-based; 2) point-based representation; 3) mesh-based; and 4) implicit neural. For each of the categories, we listed some well-known methods, explaining their contributions, challenges, strengths, and weaknesses.

Although 3-D deep surface reconstruction has made impressive progress over the last few years, there are several remaining challenges. The following nonexhaustive list will highlight the major open issues:

1) computation time;
2) generalizability;
3) energy consumption and environmental impact;
4) representation compression;
5) resolution;
6) water tightness;
7) nonrigid, dynamic, or transparent object reconstruction;
8) reconstruction of rarely visible or occluded regions, shadows, and holes in an object or a scene.

Toward the end of this article, we discussed current challenges and possible future trends in deep 3-D surface reconstruction. We assume that coming research will put a strong emphasis on self-attention-based models due to their exceling performance in DL in general and 2-D computer vision problems, i.e., vision transformer and its derivatives, in particular. Moreover, self-supervision will be the strong community focus due to its ability to not only improve reconstructive performance overall but also to leverage small and potentially domain-specific datasets. The application of diffusion models seems to be a promising direction as well. Finally, albeit in a niche setting, the quantification of reconstruction uncertainties will be of utmost importance for safety-critical applications and certain scientific application settings. ■

Acknowledgment

The authors thank their funding agencies.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[3] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[4] L. Chen, S. Peng, and X. Zhou, "Towards efficient and photorealistic 3D human reconstruction: A brief survey," *Vis. Informat.*, vol. 5, no. 4, pp. 11–19, Dec. 2021, doi: 10.1016/j.visinf.2021.10.003.

[5] Y. Tian, H. Zhang, Y. Liu, and L. Wang, "Recovering 3D human mesh from monocular images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 26, 2023, doi: 10.1109/TPAMI.2023.3298850.

[6] M. Götz, C. Bodenstein, and M. Riedel, "HPDBSCAN," in *Proc. Workshop Mach. Learn. High-Perform. Comput. Environ.*, Nov. 2015, pp. 1–10, doi: 10.1145/2834892.2834894.

[7] M. Götz, G. Cavallaro, T. Géraud, M. Book, and M. Riedel, "Parallel computation of component trees on distributed memory machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 11, pp. 2582–2598, Nov. 2018, doi: 10.1109/TPDS.2018.2829724.

[8] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. Eurographics Symp. Geometry Process.*, vol. 7, pp. 61–70, Jun. 2006, doi: 10.2312/SGP/SGP06/061-070.

[9] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, Jun. 2013, doi: 10.1145/2487228.2487237.

[10] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Trans. Vis. Comput. Graphics*, vol. 5, no. 4, pp. 349–359, Dec. 1999, doi: 10.1109/2945.817351.

[11] J.-D. Boissonnat and B. Geiger, "Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation," in *Proc. SPIE*, vol. 1905, pp. 964–975, Jul. 1993, doi: 10.1117/12.148710.

[12] S. Fortune, "Voronoi diagrams and Delaunay triangulations," in *Handbook of Discrete and Computational Geometry*. New York, NY, USA: Taylor & Francis, 1995, ch. 27, pp. 225–265, doi: 10.1142/9789812831699_0007.

[13] E. Che, J. Jung, and M. Olsen, "Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review," *Sensors*, vol. 19, no. 4, p. 810, Feb. 2019, doi: 10.3390/s19040810.

[14] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021, doi: 10.1109/TPAMI.2020.3005434.

[15] L. Jiao et al., "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

[16] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3D: A review of point cloud semantic segmentation," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 38–59, Dec. 2020, doi: 10.1109/MGRS.2019.2937630.

[17] M. Berger et al., "A survey of surface reconstruction from point clouds," *Comput. Graph. Forum*, vol. 36, no. 1, pp. 301–329, Jan. 2017, doi: 10.1111/cgf.12802.

[18] M. Zollhöfer et al., "State of the art on 3D reconstruction with RGB-D cameras," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 625–652, May 2018, doi: 10.1111/cgf.13386.

[19] X.-F. Han, H. Laga, and M. Bennamoun, "Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1578–1604, May 2021, doi: 10.1109/TPAMI.2019.2954885.

[20] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 628–644, doi: 10.1007/978-3-319-46484-8_38.

[21] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," 2016, *arXiv:1608.04236*.

[22] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 484–499, doi: 10.1007/978-3-319-46466-4_29.

[23] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," 2016, *arXiv:1610.07584*.

[24] A. Dai, C. R. Qi, and M. Nießner, "Shape completion using 3D-encoder-predictor CNNs and shape synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6545–6554, doi: 10.1109/CVPR.2017.693.

[25] D. Stutz and A. Geiger, "Learning 3D shape completion from laser scan data with weak supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1955–1964, doi: 10.1109/CVPR.2018.00209.

[26] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "OctNetFusion: Learning depth fusion from data," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 57–66, doi: 10.1109/3DV.2017.00017.

[27] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong, "Adaptive O-CNN: A patch-based deep representation of 3D shapes," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–11, Dec. 2018, doi: 10.1145/3272127.3275050.

[28] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 412–420, doi: 10.1109/3DV.2017.00054.

[29] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2107–2115, doi: 10.1109/ICCV.2017.230.

[30] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 605–613, doi: 10.1109/CVPR.2017.264.

[31] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, pp. 40–49, Jul. 2018. [Online]. Available: http://proceedings.mlr.press/v80/achlioptas18a.html

[32] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215, doi: 10.1109/CVPR.2018.00029.

[33] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "PointFlow: 3D point cloud generation with continuous normalizing flows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4540–4549, doi: 10.1109/ICCV.2019.00464.

[34] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mache approach to learning 3D surface generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 216–224, doi: 10.1109/CVPR.2018.00030.

[35] M. Liu, X. Zhang, and H. Su, "Meshing point clouds with predicted intrinsic-extrinsic ratio guidance," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 68–84, doi: 10.1007/978-3-030-58598-3_5.

[36] R. Daroya, R. Atienza, and R. Cajote, "REIN: Flexible mesh generation from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 352–353, doi: 10.1109/CVPRW50498.2020.00184.

[37] A. Badki, O. Gallo, J. Kautz, and P. Sen, "Meshlet priors for 3D mesh reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2846–2855, doi: 10.1109/CVPR42600.2020.00292.

[38] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: Generating 3D mesh models from single RGB images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2018, pp. 52–67, doi: 10.1007/978-3-030-01252-6_4.

[39] C. Wen, Y. Zhang, Z. Li, and Y. Fu, "Pixel2Mesh++: Multi-view 3D mesh generation via deformation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1042–1051, doi: 10.1109/ICCV.2019.00113.

[40] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 371–386, doi: 10.1007/978-3-030-01267-0_23.

[41] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, "Point2Mesh: A self-prior for deformable meshes," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 1–12, Aug. 2020, doi: 10.1145/3386569.3392415.

[42] Y. Liao, S. Donné, and A. Geiger, "Deep marching cubes: Learning explicit surface representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2916–2925, doi: 10.1109/CVPR.2018.00308.

[43] A. Dai and M. NieBner, "Scan2Mesh: From unstructured range scans to 3D meshes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5574–5583, doi: 10.1109/CVPR.2019.00572.

[44] G. Gkioxari, J. Johnson, and J. Malik, "Mesh R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9784–9794, doi: 10.1109/ICCV.2019.00988.

[45] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4460–4470, doi: 10.1109/CVPR.2019.00459.

[46] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-58580-8_31.

[47] R. Chabra et al., "Deep local shapes: Learning local SDF priors for detailed 3D reconstruction," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 608–625, doi: 10.1007/978-3-030-58526-6_36).

[48] P. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer, "POINTS2SURF learning implicit surfaces from point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 108–124, doi: 10.1007/978-3-030-58558-7_7.

[49] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5939–5948, doi: 10.1109/CVPR.2019.00609.

[50] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 165–174, doi: 10.1109/CVPR.2019.00025.

[51] V. Sitzmann, J. Martel, A. Bergman, D. Lindell,

and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 7462–7473. [Online]. Available: https://proceedings.neurips.cc//paper/2020/file/53c04118df112c13a8c34b38343b9c10-Paper.pdf

[52] M. Atzmon and Y. Lipman, "SAL: Sign agnostic learning of shapes from raw data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2562–2571, doi: 10.1109/CVPR42600.2020.00264.

[53] J. Chibane, A. Mir, and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," 2020, *arXiv:2010.13938*.

[54] R. Venkatesh, S. Sharma, A. Ghosh, L. Jeni, and M. Singh, "DUDE: Deep unsigned distance embeddings for hi-fidelity representation of complex 3D surfaces," 2020, *arXiv:2011.02570*.

[55] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, and T. Funkhouser, "Local implicit grid representations for 3D scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6000–6009, doi: 10.1109/CVPR42600.2020.00604.

[56] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3D shape reconstruction and completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6970–6981, doi: 10.1109/CVPR42600.2020.00700.

[57] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 405–421, doi: 10.1007/978-3-030-58452-8_24.

[58] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "FastNeRF: High-fidelity neural rendering at 200FPS," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14326–14335, doi: 10.1109/ICCV48922.2021.01408.

[59] T. Neff et al., "DONeRF: Towards real-time rendering of compact neural radiance fields using depth Oracle networks," *Comput. Graph. Forum*, vol. 40, no. 4, pp. 45–59, Jul. 2021, doi: 10.1111/cgf.14340.

[60] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14315–14325, doi: 10.1109/ICCV48922.2021.01407.

[61] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "NeRF++: Analyzing and improving neural radiance fields," 2020, *arXiv:2010.07492*.

[62] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "PixelNeRF: Neural radiance fields from one or few images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4576–4585, doi: 10.1109/CVPR46437.2021.00455.

[63] A. Chen et al., "MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14104–14113, doi: 10.1109/ICCV48922.2021.01386.

[64] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "NeRV: Neural reflectance and visibility fields for relighting and view synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7491–7500, doi: 10.1109/CVPR46437.2021.00741.

[65] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. A. Lensch, "NeRD: Neural reflectance decomposition from image collections," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12664–12674, doi: 10.1109/ICCV48922.2021.01245.

[66] G. Gafni, J. Thies, M. Zollhöfer, and M. Nießner, "Dynamic neural radiance fields for monocular 4D facial avatar reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8645–8654, doi: 10.1109/CVPR46437.2021.00854.

[67] K. Park et al., "Nerfies: Deformable neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5845–5854, doi: 10.1109/ICCV48922.2021.00581.

[68] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt, "Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12939–12950, doi: 10.1109/ICCV48922.2021.01272.

[69] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D mesh renderer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3907–3916, doi: 10.1109/CVPR.2018.00411.

[70] J. Pan, X. Han, W. Chen, J. Tang, and K. Jia, "Deep mesh reconstruction from single RGB images via topology modification networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9963–9972, doi: 10.1109/ICCV.2019.01006.

[71] J. Tang, X. Han, J. Pan, K. Jia, and X. Tong, "A skeleton-bridged deep learning approach for generating meshes of complex topologies from single RGB images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4536–4545, doi: 10.1109/CVPR.2019.00467.

[72] Q. Wang, Y. Tan, and Z. Mei, "Computational methods of acquisition and processing of 3D point cloud data for construction applications," *Arch. Comput. Methods Eng.*, vol. 27, no. 2, pp. 479–499, Apr. 2020, doi: 10.1007/s11831-019-09320-4.

[73] J. Shan and C. K. Toth, *Topographic Laser Ranging and Scanning: Principles and Processing*. Boca Raton, FL, USA: CRC Press, 2017, doi: 10.1201/9781315154381.

[74] Y. Gu, Q. Wang, X. Jia, and J. A. Benediktsson, "A novel MKL model of integrating LiDAR data and MSI for urban area classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5312–5326, Oct. 2015, doi: 10.1109/TGRS.2015.2421051.

[75] J. Fernandez-Diaz et al., "Capability assessment and performance metrics for the Titan multispectral mapping LiDAR," *Remote Sens.*, vol. 8, no. 11, p. 936, Nov. 2016, doi: 10.3390/rs8110936. [Online]. Available: https://www.mdpi.com/2072-4292/8/11/936

[76] M. Pedergnana, P. R. Marpu, M. D. Mura, J. A. Benediktsson, and L. Bruzzone, "Classification of remote sensing optical and LiDAR data using extended attribute profiles," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 7, pp. 856–865, Nov. 2012, doi: 10.1109/JSTSP2012.2208177.

[77] M. Kukkonen, M. Maltamo, L. Korhonen, and P. Packalen, "Comparison of multispectral airborne laser scanning and stereo matching of aerial images as a single sensor solution to forest inventories by tree species," *Remote Sens. Environ.*, vol. 231, Sep. 2019, Art. no. 111208, doi: 10.1016/j.rse.2019.05.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425719302214

[78] M. A. Isa and I. Lazoglu, "Design and analysis of a 3D laser scanner," *Measurement*, vol. 111, pp. 122–133, Dec. 2017, doi: 10.1016/j.measurement.2017.07.028. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224117304633

[79] F. Rottensteiner et al., "The ISPRS benchmark on urban object classification and 3D building reconstruction," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. I-3, pp. 293–298, Jul. 2012, doi: 10.5194/isprsannals-i-3-293-2012.

[80] Wang, Chen, Zhu, Liu, Li, and Zheng, "A survey of mobile laser scanning applications and key techniques over urban areas," *Remote Sens.*, vol. 11, no. 13, p. 1540, Jun. 2019, doi: 10.3390/rs11131540. [Online]. Available: https://www.mdpi.com/2072-4292/11/13/1540

[81] M. Elhousni and X. Huang, "A survey on 3D LiDAR localization for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1879–1884, doi: 10.1109/IV47402.2020.9304812.

[82] J. Li, B. Yang, Y. Cong, L. Cao, X. Fu, and Z. Dong, "3D forest mapping using a low-cost UAV laser scanning system: Investigation and comparison," *Remote Sens.*, vol. 11, no. 6, p. 717, Mar. 2019, doi: 10.3390/rs11060717.

[83] T. Zwęgliński, "The use of drones in disaster aerial needs reconnaissance and damage assessment—Three-dimensional modeling and orthophoto map study," *Sustainability*, vol. 12, no. 15, p. 6080, Jul. 2020, doi: 10.3390/su12156080. https://www.mdpi.com/2071-1050/12/15/6080

[84] F. Leberl et al., "Point clouds: LiDAR versus 3D vision," *Photogramm. Eng. Remote Sens.*, vol. 76, no. 10, pp. 1123–1134, Oct. 2010, doi: 10.14358/PERS.76.10.1123.

[85] Q. Hu, J. Luo, G. Hu, W. Duan, and H. Zhou, "3D point cloud generation using incremental structure-from-motion," *J. Phys., Conf. Ser.*, vol. 1087, Sep. 2018, Art. no. 062031, doi: 10.1088/1742-6596/1087/6/062031.

[86] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 519–528, doi: 10.1109/CVPR.2006.19.

[87] K. Kamal et al., "Performance assessment of Kinect as a sensor for pothole imaging and metrology," *Int. J. Pavement Eng.*, vol. 19, no. 7, pp. 565–576, Jul. 2018, doi: 10.1080/10298436.2016.1187730.

[88] C. Jia, T. Yang, C. Wang, B. Fan, and F. He, "A new fast filtering algorithm for a 3D point cloud based on RGB-D information," *PLoS ONE*, vol. 14, no. 8, Aug. 2019, Art. no. e0220253, doi: 10.1371/journal.pone.0220253.

[89] C. Chen, B. S. Yang, and S. Song, "Low cost and efficient 3D indoor mapping using multiple consumer RGB-D cameras," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLI-B1, pp. 169–174, Jun. 2016, doi: 10.5194/isprs-archives-XLI-B1-169-2016.

[90] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight Kinect," *Comput. Vis. Image Understand.*, vol. 139, pp. 1–20, Oct. 2015, doi: 10.1016/j.cviu.2015.05.006.

[91] R. Bürgmann, P. A. Rosen, and E. J. Fielding, "Synthetic aperture radar interferometry to measure Earth's surface topography and its deformation," *Annu. Rev. Earth Planet. Sci.*, vol. 28, no. 1, pp. 169–209, May 2000, doi: 10.1146/annurev.earth.28.1.169.

[92] A. Ferretti, C. Prati, and F. Rocca, "Permanent scatterers in SAR interferometry," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jun. 1999, pp. 1528–1530, doi: 10.1109/IGARSS.1999.772008.

[93] A. Budillon, M. Crosetto, and O. Monserrat, "Editorial for the special issue 'urban deformation monitoring using persistent scatterer interferometry and SAR tomography,'" *Remote Sens.*, vol. 11, no. 11, p. 1306, May 2019, doi: 10.3390/rs11111306. https://www.mdpi.com/2072-4292/11/11/1306

[94] A. Gruen, "Fundamentals of videogrammetry—A review," *Hum. Movement Sci.*, vol. 16, no. 2, pp. 155–187, 1997, doi: 10.1016/S0167-9457(96)00048-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167945796000486

[95] A. Torresani and F. Remondino, "Videogrammetry VS photogrammetry for heritage 3D reconstruction," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W15, pp. 1157–1162, Aug. 2019, doi: 10.5194/isprs-archives-XLII-2-W15-1157-2019.

[96] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[97] K. Mo et al., "PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object

understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 909–918, doi: 10.1109/CVPR.2019.00100.

[98] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920, doi: 10.1109/CVPR.2015.7298801.

[99] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3354–3361, Jun. 2012, doi: 10.1109/CVPR.2012.6248074. [Online]. Available: http://www.cvlibs.net/datasets/kitti/index.php

[100] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013, doi: 10.1177/0278364913491297. [Online]. Available: http://www.cvlibs.net/datasets/kitti/index.php

[101] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9297–9307, doi: 10.1109/ICCV.2019.00939. [Online]. Available: http://www.semantic-kitti.org/

[102] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2432–2443, doi: 10.1109/CVPR.2017.261.

[103] A. Chang et al., "Matterport3D: Learning from RGB-D data in indoor environments," in *Proc. Int. Conf. 3D Vis. (3DV)*, pp. 667–676, Oct. 2017, doi: 10.1109/3DV.2017.00081. [Online]. Available: https://niessner.github.io/Matterport/

[104] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Eur. Conf. Comput. Vis.* Springer, 2012, pp. 746–760, doi: 10.1007/978-.3-642-33715-4_54. [Online]. Available: https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

[105] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1625–1632, doi: 10.1109/ICCV.2013.458. [Online]. Available: http://sun3d.cs.princeton.edu/

[106] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 567–576, doi: 10.1109/CVPR.2015.7298655.

[107] A. Janoch et al., "A category-level 3-D object dataset: Putting the Kinect to work," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 141–165, doi: 10.1109/ICCVW.2011.6130382.

[108] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, "Unsupervised feature learning for classification of outdoor 3D scans," in *Proc. Australas. Conf. Robot. Autom.*, vol. 2, 2013, p. 1. [Online]. Available: http://www.acfr.usyd.edu.au/papers/SydneyUrbanObjectsDataset.shtml

[109] S. Koch et al., "ABC: A big CAD model dataset for geometric deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9593–9603, Jun. 2019, doi: 10.1109/CVPR.2019.00983. [Online]. Available: https://cs.nyu.edu/~zhongshi/publication/abc-dataset/

[110] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D.Net: A new large-scale point cloud classification benchmark," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. IV-1-W1, pp. 91–98, May 2017, doi: 10.5194/ISPRS-ANNALS-IV-1-W1-91-2017. [Online]. Available: https://www.semantic3d.net/view_dbase.php?chl=1&orderBy=name&orderStyle=ASC#download

[111] M. Kölle et al., "The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo," 2021, *arXiv:2102.05346*.

[112] H. Fu et al., "3D-FRONT: 3D furnished rooms with layOuts and semaNTics," 2020, *arXiv:2011.09127*.

[113] H. Fu et al., "3D-FUTURE: 3D furniture shape with TextURE," 2020, *arXiv:2009.09633*.

[114] Q. Hu, B. Yang, S. Khalid, W. Xiao, N. Trigoni, and A. Markham, "Towards semantic segmentation of urban-scale 3D point clouds: A dataset, benchmarks and challenges," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4975–4985. [Online]. Available: http://point-cloud-analysis.cs.ox.ac.uk/

[115] (2014). *The Stanford Computer Graphics Laboratory*. [Online]. Available: https://graphics.stanford.edu/data/3Dscanrep/

[116] G. Turk and B. Mullins. (2021). *Large Geometric Models Archive*. [Online]. Available: https://www.cc.gatech.edu/projects/large_models/

[117] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 59–66, doi: 10.1109/ICCV.1998.710701.

[118] Y. Rubner, "The Earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000, doi: 10.1023/A:1026543900054.

[119] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3D reconstruction networks learn?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3400–3409, doi: 10.1109/CVPR.2019.00352.

[120] O. Taubert, M. Götz, A. Schug, and A. Streit, "Loss scheduling for class-imbalanced image segmentation problems," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2020, pp. 426–431, doi: 10.1109/ICMLA51294.2020.00073.

[121] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951, doi: 10.1214/AOMS/1177729694.

[122] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, Sep. 2003, doi: 10.1111/1467-8659.00669.

[123] H. Kato et al., "Differentiable rendering: A survey," 2020, *arXiv:2006.12057*.

[124] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, Aug. 1996, pp. 303–312, doi: 10.1145/237170.237269.

[125] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *Proc. 14th Annu. Conf. Comput. Graph. Interact. Techn.*, Aug. 1987, pp. 163–169, doi: 10.1145/37401.37422.

[126] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2015, pp. 922–928, doi: 10.1109/IROS.2015.7353481.

[127] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3D object recognition," 2016, *arXiv:1604.03351*.

[128] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656, doi: 10.1109/CVPR.2016.609.

[129] V. Hegde and R. Zadeh, "FusionNet: 3D object classification using multiple data representations," 2016, *arXiv:1607.05695*.

[130] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Symp. Geometry Process.*, vol. 6, pp. 156–164, doi: 10.2312/SGP/SGP03/156-165.

[131] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953, doi: 10.1109/ICCV.2015.114.

[132] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[133] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Category-specific object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1966–1974, doi: 10.1109/CVPR.2015.7298807.

[134] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[135] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019, doi: 10.1561/2200000056.

[136] I. J. Goodfellow et al., "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[137] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[138] J. Wu, C. Zhang, X. Zhang, Z. Zhang, W. T. Freeman, and J. B. Tenenbaum, "Learning shape priors for single-view 3D completion and reconstruction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 646–662, doi: 10.1007/978-3-030-01252-6_40.

[139] A. Dai, C. Diller, and M. Niessner, "SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 846–855, doi: 10.1109/cvpr42600.2020.00093.

[140] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal ConvNets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3070–3079, doi: 10.1109/CVPR.2019.00319.

[141] S. Roth and S. R. Richter, "Matryoshka networks: Predicting 3D geometry via nested shape layers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1936–1944, doi: 10.1109/CVPR.2018.00207.

[142] D. Meagher, "Geometric modeling using Octree encoding," *Comput. Graph. Image Process.*, vol. 19, no. 2, pp. 129–147, Jun. 1982, doi: 10.1016/0146-664X(82)90104-6.

[143] C. L. Jackins and S. L. Tanimoto, "Oct-trees and their use in representing three-dimensional objects," *Comput. Graph. Image Process.*, vol. 14, no. 3, pp. 249–270, Nov. 1980, doi: 10.1016/0146-664X(80)90055-6.

[144] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3577–3586, doi: 10.1109/CVPR.2017.701.

[145] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust TV-L^1 range image integration," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8, doi: 10.1109/ICCV.2007.4408983.

[146] M. Firman, O. M. Aodha, S. Julier, and G. J. Brostow, "Structured prediction of unobserved voxels from a single depth image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5431–5440, doi: 10.1109/CVPR.2016.586.

[147] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Aug. 2017, doi: 10.1145/3072959.3073608.

[148] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*,

Jul. 2017, pp. 652–660, doi: 10.1109/CVPR.2017.16.
[149] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31. Red Hook, NY, USA: Curran Associates, 2017, pp. 5099–5108. [Online]. Available: https://papers.nips.cc/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf
[150] R. Klokov and V. Lempitsky, "Escape from cells: Deep Kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872, doi: 10.1109/ICCV.2017.99.
[151] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4548–4557, doi: 10.1109/CVPR.2018.00478.
[152] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1530–1538. [Online]. Available: http://proceedings.mlr.press/v37/rezende15.html
[153] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," 2018, *arXiv:1806.07366*.
[154] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, "FFJORD: Free-form continuous dynamics for scalable reversible generative models," 2018, *arXiv:1810.01367*.
[155] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-Net: Point cloud upsampling network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2790–2799, doi: 10.1109/CVPR.2018.00295.
[156] W. Yifan, S. Wu, H. Huang, D. Cohen-Or, and O. Sorkine-Hornung, "Patch-based progressive 3D point set upsampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5951–5960, doi: 10.1109/CVPR.2019.00611.
[157] R. Cai, G. Yang, H. Averbuch-Elor, Z. Hao, S. Belongie, N. Snavely, and B. Hariharan, "Learning gradient fields for shape generation," in *Proc. Eur. Conf. Comput. Vis.*, vol. 12348, 2020, pp. 364–381, doi: 10.1007/978-3-030-58580-8_22.
[158] P. Guerrero, Y. Kleiman, M. Ovsjanikov, and N. J. Mitra, "PCPNet learning local shape properties from raw point clouds," *Comput. Graph. Forum*, vol. 37, pp. 75–85, May 2018, doi: 10.1111/cgf.13343.
[159] Y. Ben-Shabat and S. Gould, "DeepFit: 3D surface fitting via neural network weighted least squares," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 20–34, doi: 10.1007/978-3-030-58452-8_2.
[160] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–12, Aug. 2018, doi: 10.1145/3197517.3201301.
[161] J. Mao, X. Wang, and H. Li, "Interpolated convolutional networks for 3D point cloud understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1578–1587, doi: 10.1109/ICCV.2019.00166.
[162] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5560–5568, doi: 10.1109/CVPR.2019.00571.
[163] S. Lan, R. Yu, G. Yu, and L. S. Davis, "Modeling local geometric structure of 3D point clouds using geo-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 998–1008, doi: 10.1109/CVPR.2019.00109.
[164] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9613–9622, doi: 10.1109/CVPR.2019.00985.
[165] Y. Zhao, T. Birdal, J. E. Lenssen, E. Menegatti, L. Guibas, and F. Tombari, "Quaternion equivariant capsule networks for 3D point clouds," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 1–19, doi: 10.1007/978-3-030-58452-8_1.
[166] F. Engelmann, T. Kontogianni, and B. Leibe, "Dilated point convolutions: On the receptive field size of point convolutions on 3D point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9463–9469, doi: 10.1109/ICRA40945.2020.9197503.
[167] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou, "Tangent convolutions for dense prediction in 3D," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3887–3896, doi: 10.1109/CVPR.2018.00409.
[168] H. Su et al., "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539, doi: 10.1109/CVPR.2018.00268.
[169] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10526–10535, doi: 10.1109/cvpr42600.2020.01054.
[170] M. Rakotosaona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, "PointCleanNet: Learning to denoise and remove outliers from dense point clouds," *Comput. Graph. Forum*, vol. 39, no. 1, pp. 185–203, Feb. 2020, doi: 10.1111/cgf.13753.
[171] E. J. Smith, S. Fujimoto, A. Romero, and D. Meger, "GEOMetrics: Exploiting geometric structure for graph-encoded objects," 2019, *arXiv:1901.11461*.
[172] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson, and C. Fookes, "Image2Mesh: A learning framework for single image 3D reconstruction," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 365–381, doi: 10.1007/978-3-030-20887-5_23.
[173] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/cvpr.2016.90.
[174] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, "Deep geometric prior for surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10122–10131, doi: 10.1109/CVPR.2019.01037.
[175] A. Kar, C. Häne, and J. Malik, "Learning a multi-view stereo machine," 2017, *arXiv:1708.05375*.
[176] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "MeshCNN: A network with an edge," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Aug. 2019, doi: 10.1145/3306346.3322959.
[177] D. Coquelin et al., "Accelerating neural network training with distributed asynchronous and selective optimization (DASO)," *J. Big Data*, vol. 9, no. 1, pp. 1–18, Dec. 2022, doi: 10.1186/s40537-021-00556-1.
[178] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.
[179] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "GraphRNN: Generating realistic graphs with deep auto-regressive models," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 5708–5717. [Online]. Available: http://proceedings.mlr.press/v80/you18a/you18a.pdf
[180] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1747–1756. [Online]. Available: http://proceedings.mlr.press/v48/oord16.pdf
[181] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989, doi: 10.1016/0893-6080(89)90020-8.
[182] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. 19th Annu. Conf. Comput. Graph. Interact. Techn.*, Jul. 1992, pp. 71–78, doi: 10.1145/133994.134011.
[183] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Stoll, and C. Theobalt, "PatchNets: Patch-based generalizable deep implicit 3D shape representations," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 293–309, doi: 10.1007/978-3-030-58517-4_18.
[184] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, "Local deep implicit functions for 3D shape," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4856–4865, doi: 10.1109/CVPR42600.2020.00491.
[185] J. Chibane and G. Pons-Moll, "Implicit feature networks for texture completion from partial 3D data," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 717–725, doi: 10.1007/978-3-030-66096-3_48.
[186] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.
[187] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2016, pp. 424–432, doi: 10.1007/978-3-319-46723-8_49.
[188] S. Saito, T. Simon, J. Saragih, and H. Joo, "PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 81–90, doi: 10.1109/cvpr42600.2020.00016.
[189] B. Lal Bhatnagar, C. Sminchisescu, C. Theobalt, and G. Pons-Moll, "Combining implicit function learning and parametric models for 3D human reconstruction," 2020, *arXiv:2007.11432*.
[190] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Occupancy flow: 4D reconstruction by learning particle dynamics," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5378–5388, doi: 10.1109/ICCV.2019.00548.
[191] S. Saito, Z. Huang, R. Natsume, S. Morishima, H. Li, and A. Kanazawa, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2304–2314, doi: 10.1109/ICCV.2019.00239.
[192] M. Oechsle, L. Mescheder, M. Niemeyer, T. Strauss, and A. Geiger, "Texture fields: Learning texture representations in function space," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4530–4539, doi: 10.1109/ICCV.2019.00463.
[193] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3D-Structure-Aware neural scene representations," 2019, *arXiv:1906.01618*.
[194] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3501–3512, doi: 10.1109/CVPR42600.2020.00356.
[195] L. Yariv et al., "Multiview neural surface reconstruction by disentangling geometry and appearance," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 2492–2502. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/1a77befc3b608d6ed363567685f70e1e-Paper.pdf
[196] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann, "DISN: Deep implicit surface network for high-quality single-view 3D reconstruction," 2019, *arXiv:1905.10711*.
[197] S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, and

Z. Cui, "DIST: Rendering deep implicit signed distance function with differentiable sphere tracing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2016–2025, doi: 10.1109/CVPR42600.2020.00209.

[198] Y. Jiang, D. Ji, Z. Han, and M. Zwicker, "SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1248–1258, doi: 10.1109/cvpr42600.2020.00133.

[199] Y. Duan, H. Zhu, H. Wang, L. Yi, R. Nevatia, and L. J. Guibas, "Curriculum DeepSDF," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 51–67, doi: 10.1007/978-3-030-58598-3_4.

[200] T. Takikawa et al., "Neural geometric level of detail: Real-time rendering with implicit 3D shapes," 2021, *arXiv:2101.10994*.

[201] S. Duggal et al., "Mending neural implicit modeling for 3D vehicle reconstruction in the wild," 2021, *arXiv:2101.06860*.

[202] S. Liu, S. Saito, W. Chen, and H. Li, "Learning to infer implicit surfaces without 3D supervision," 2019, *arXiv:1911.00767*.

[203] E. Chatzipantazis, S. Pertigkiozoglou, E. Dobriban, and K. Daniilidis, "SE(3)-equivariant attention networks for shape reconstruction in function space," 2022, *arXiv:2204.02394*.

[204] F. Fuchs, D. Worrall, V. Fischer, and M. Welling, "SE(3)-transformers: 3D roto-translation equivariant attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1970–1981.

[205] N. Thomas et al., "Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds," 2018, *arXiv:1802.08219*.

[206] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. Guibas, "Vector neurons: A general framework for SO(3)-equivariant networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12180–12189, doi: 10.1109/ICCV48922.2021.01198.

[207] Y. Chen, B. Fernando, H. Bilen, M. Nießner, and E. Gavves, "3D equivariant graph implicit functions," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 485–502.

[208] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008. [Online]. Available: https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[209] Q. Wang et al., "IBRNet: Learning multi-view image-based rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4688–4697, doi: 10.1109/CVPR46437.2021.00466.

[210] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5835–5844, doi: 10.1109/ICCV48922.2021.00580.

[211] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded anti-aliased neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 5470–5479, doi: 10.1109/CVPR52688.2022.00539.

[212] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Zip-NeRF: Anti-aliased grid-based neural radiance fields," 2023, *arXiv:2304.06706*.

[213] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–15, Jul. 2022, doi: 10.1145/3528223.3530127.

[214] S. Bi et al., "Neural reflectance fields for appearance acquisition," 2020, *arXiv:2008.03824*.

[215] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "NeRFactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 1–18, Dec. 2021, doi: 10.1145/3478513.3480496.

[216] Z. Wang, S. Wu, W. Xie, M. Chen, and V. Adrian Prisacariu, "NeRF–: Neural radiance fields without known camera parameters," 2021, *arXiv:2102.07064*.

[217] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural radiance fields for dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10313–10322, doi: 10.1109/CVPR46437.2021.01018.

[218] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6494–6504, doi: 10.1109/CVPR46437.2021.00643.

[219] K. Park et al., "HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields," 2021, *arXiv:2106.13228*.

[220] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5855–5864, doi: 10.1109/ICCV48922.2021.00582.

[221] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, "NeRF: Neural radiance field in 3D vision, a comprehensive review," 2022, *arXiv:2210.00379*.

[222] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 27171–27183. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/e41e164f7485ec4a28741a2d0ea41c74-Paper.pdf

[223] M. Oechsle, S. Peng, and A. Geiger, "UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5569–5579.

[224] Y. Wang, I. Skorokhodov, and P. Wonka, "HF-NeuS: Improved surface reconstruction using high-frequency details," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 1966–1978. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/0ce8e3434c7b486bbddff9745b2a1722-Paper-Conference.pdf

[225] Q. Fu, Q. Xu, Y. S. Ong, and W. Tao, "Geo-Neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 3403–3416. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/16415eed5a0a121bfce79924db05d3fe-Paper-Conference.pdf

[226] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 4805–4815. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/25e2a30f44898b9f3e978b1786dcd85c-Paper.pdf

[227] Z. Yu et al. (2022). *SDFStudio: A Unified Framework for Surface Reconstruction*. [Online]. Available: https://github.com/autonomousvision/sdfstudio

[228] M. Tancik et al., "Nerfstudio: A modular framework for neural radiance field development," in *Proc. Special Interest Group Comput. Graph. Interact. Techn. Conf. Conf.*, Jul. 2023, pp. 1–12, doi: 10.1145/3588432.3591516.

[229] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond Pascal: A benchmark for 3D object detection in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 75–82, doi: 10.1109/WACV.2014.6836101.

[230] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4004–4012, doi: 10.1109/CVPR.2016.434.

[231] J. J. Lim, H. Pirsiavash, and A. Torralba, "Parsing IKEA objects: Fine pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2992–2999, doi: 10.1109/ICCV.2013.372.

[232] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun, "A large dataset of object scans," 2016, *arXiv:1602.02481*.

[233] Trimble Inc. (2023). *3D Warehouse*. [Online]. Available: https://3dwarehouse.sketchup.com/

[234] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, "Large scale multi-view stereopsis evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 406–413.

[235] Y. Yao et al., "BlendedMVS: A large-scale dataset for generalized multi-view stereo networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1787–1796.

[236] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "SceneNet RGB-D: Can 5M synthetic images beat generic ImageNet pre-training on indoor segmentation?" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2697–2706, doi: 10.1109/ICCV.2017.292.

[237] A. Bozic, P. Palafox, J. Thies, A. Dai, and M. Niessner, "TransformerFusion: Monocular RGB scene reconstruction using transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2021, pp. 1403–1414. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/0a87257e5308197df43230edf4ad1dae-Paper.pdf

[238] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 16239–16248, Oct. 2021. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/papers/Zhao_Point_Transformer_ICCV_2021_paper.pdf

[239] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, Jun. 2021, doi: 10.1007/s41095-021-0229-5.

[240] Y. LeCun and I. Misra. (2021). *Self-Supervised Learning: The Dark Matter of Intelligence*. [Online]. Available: https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/

[241] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 2256–2265. [Online]. Available: http://proceedings.mlr.press/v37/sohl-dickstein15.pdf

[242] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[243] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8162–8171. [Online]. Available: http://proceedings.mlr.press/v139/nichol21a/nichol21a.pdf

[244] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8780–8794. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf

[245] S. Luo and W. Hu, "Diffusion probabilistic models for 3D point cloud generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2836–2844, doi: 10.1109/CVPR46437.2021.00286.

[246] L. Zhou, Y. Du, and J. Wu, "3D shape generation and completion through point-voxel diffusion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5806–5815, doi: 10.1109/ICCV48922.2021.00577.

ABOUT THE AUTHORS

Anis Farshian received the B.Sc. degree in computer engineering from the Technical Faculty, Shariati Technical and Vocational College, Tehran, Iran, in 2012, and the M.Sc. degree in information technology engineering from the Iran University of Science and Technology, Tehran, in 2017.

She is currently with the Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany. As part of her research and her membership in the Helmholtz Analytics Framework project and the Helmholtz AI project, she is working on methods for surface reconstruction for applied scientific problems. Her research interests include 3-D analysis, machine learning, and computer-assisted health.

Markus Götz (Member, IEEE) received the B.Sc. and M.Sc. degrees in IT-system engineering from the University of Potsdam, Potsdam, Germany, in 2010 and 2014, respectively, with intermediate stays at the Blekinge Tekniska Högskola, Karlskrona, Sweden, and CERN, Meyrin, Switzerland. He is currently working toward the Ph.D. degree in computational engineering at the University of Iceland, Reykjavik, Iceland, in conjunction with the Juelich Supercomputing Centre, Jülich, Germany.

He is currently a Postdoctoral Researcher with the Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany, where he is also the Project Manager for the Helmholtz Analytics Framework and the Head of the Helmholtz AI Consultants Team. In line with his work, he focuses on applied artificial intelligence and data analysis on high-performance cluster systems to work on the grand challenges in the natural sciences. His research interests include machine learning, global optimization, and parallel algorithm engineering.

Gabriele Cavallaro (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunications engineering from the University of Trento, Trento, Italy, in 2011 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2016.

From 2016 to 2021, he was the Deputy Head of the High Productivity Data Processing (HPDP) Research Group, Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich, Jülich, Germany. Since 2022, he has been the Head of the "AI and ML for Remote Sensing" Simulation and Data Laboratory, JSC, and an Adjunct Associate Professor with the School of Natural Sciences and Engineering, University of Iceland. Concurrently, he serves as a Visiting Professor at the Φ-Laboratory, European Space Agency (ESA), Italy, where he contributes to the Quantum Computing for Earth Observation (QC4EO) initiative. His research interests cover remote sensing data processing with parallel machine learning algorithms that scale on distributed computing systems and innovative computing technologies.

Dr. Cavallaro was a recipient of the IEEE GRSS Third Prize in the Student Paper Competition of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2015 (Milan, Italy). From 2020 to 2023, he was the Chair of the High-Performance and Disruptive Computing in Remote Sensing (HDCRS) Working Group under the IEEE GRSS Earth Science Informatics Technical Committee (ESI TC). In 2023, he took on the role of Co-Chair of the ESI TC. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP) since October 2022.

Charlotte Debus (Member, IEEE) studied physics at the University of Heidelberg, Heidelberg, Germany. She received the M.Sc. degree from the University of Heidelberg in 2012 and the Ph.D. degree in medical physics from the German Cancer Research Center (DKFZ), Heidelberg, in 2016.

After a two-year postdoctoral period at DKFZ, she was a Research Associate with the German Aerospace Institute, Cologne, Germany, from 2019 to 2020, where she worked on machine learning methods and high-performance computing. Since October 2020, she has been a Postdoctoral Researcher at the Helmholtz AI Local Energy Consultants Team, Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany.

Matthias Nießner received the Ph.D. degree from the University of Erlangen-Nuremberg, Erlangen, Germany, in 2013.

He was a Visiting Assistant Professor with Stanford University, Stanford, CA, USA, from 2013 to 2017. Since 2017, he has been a Professor with the Technical University of Munich (TUM), Munich, Germany, focusing on cutting-edge research at the intersection of computer vision, graphics, and machine learning. He is currently the Head of the Visual Computing Laboratory, TUM. In addition to his academic career, he is a Co-Founder and the Director of Synthesia Inc., London, U.K., a startup empowering storytellers with artificial intelligence (AI). He is particularly interested in novel techniques for 3-D reconstruction, semantic 3-D scene understanding, and video editing.

Prof. Nießner is a TUM-IAS Rudolph Moessbauer Fellow. He received the Google Faculty Award for Machine Perception in 2017, the Nvidia Professor Partnership Award in 2018, and the ERC Starting Grant in 2018.

Jón Atli Benediktsson (Fellow, IEEE) received the Cand.Sci. degree in electrical engineering from the University of Iceland, Reykjavik, Iceland, in 1984, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1987 and 1990, respectively.

From 2009 to 2015, he was a Pro-Rector of science and academic affairs and a Professor of electrical and computer engineering with the University of Iceland. Since July 2015, he has been the President and the Rector of the University of Iceland. He is a Co-Founder of the biomedical startup company Oxymap, Reykjavik. His research interests are in remote sensing, biomedical analysis of signals, pattern recognition, image processing, and signal processing. He has published extensively in these fields.

Dr. Benediktsson is a fellow of the Optics and Photonics Society (SPIE). He was a member of the 2014 IEEE Fellow Committee. He is a member of the Academia Europea, the Association of Chartered Engineers in Iceland (VFI), Societas Scientiarum Islandica, and Tau Beta Pi. He received the Stevan J. Kristof Award from Purdue University in 1991 as an outstanding graduate student in remote sensing. He was a recipient of the Icelandic Research Council's Outstanding Young Researcher Award in 1997.

In 2000, he was granted the IEEE Third Millennium Medal. In 2004, he was a co-recipient of the University of Iceland's Technology Innovation Award. In 2006, he received the Yearly Research Award from the Engineering Research Institute of the University of Iceland. He received the Outstanding Service Award from the IEEE Geoscience and Remote Sensing Society (GRSS) in 2007, the IEEE GRSS Education Award in 2020, the IEEE GRSS David Landgrebe Award in 2018, and the OECE Award from the School of Electrical and Computer Engineering (ECE), Purdue University, in 2016. He was co-recipient of the 2012 IEEE Transactions on Geoscience and Remote Sensing Paper Award. In 2013, he was a co-recipient of the IEEE GRSS Highest Impact Paper Award. In 2013, he received the IEEE/VFI Electrical Engineer of the Year Award. In 2016 and 2018, he was a co-recipient of the International Journal of Image and Data Fusion Best Paper Awards. In 2021, he was honored as a recipient of the Order of the Falcon from the President of Iceland. He was a Highly Cited Researcher (Clarivate Analysis) from 2018 to 2020. He was the 2011–2012 President of the IEEE GRSS and has been on the GRSS AdCom since 2000. He was the Editor-in-Chief of IEEE Transactions on Geoscience and Remote Sensing (TGRS) from 2003 to 2008. He has been serving as an Associate Editor of TGRS since 1999, IEEE Geoscience and Remote Sensing Letters since 2003, and IEEE Access since 2013. He was on the Editorial Board of the Proceedings of the IEEE from 2015 to 2020. He is on the International Editorial Board of *International Journal of Image and Data Fusion* and the Editorial Board of *Remote Sensing*. He was the Chairperson of the Steering Committee of IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS) from 2007 to 2010.

Achim Streit received the Diploma degree in computer science from the University of Dortmund, Dortmund, Germany, in 1999, and the Ph.D. degree in computer science from the University of Paderborn, Paderborn, Germany, in 2003.

He is one of the directors of the Steinbuch Centre for Computing, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. He is also a Professor of distributed and parallel high-performance computing systems with the Department of Informatics, KIT. Afterward, he led the Federated Systems and Data Division, Jülich Supercomputing Centre, Jülich, Germany. He initiated and chaired several national and international research initiatives within the Helmholtz Association [e.g., Helmholtz Data Federation and Helmholtz Information & Data Science Academy (HIDA)] on the national level (e.g., NFDI4Ing and NFDI-MatWerk) and the European level (e.g., EUDAT and EOSC). His research interests include high-performance and data-intensive computing, big data and federated data management, data analytics, job scheduling, and resource management for parallel and distributed systems.