IMPERIAL COLLEGE LONDON

DEPARTMENT OF LIFE SCIENCE

Artificial intelligence approaches to optimising segmentation in computed tomography data processing

Author:
Yuheng Wang

Supervisor:
Dr. Martin D. Brazeau

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Research at Imperial College London

Formatted in the journal style of the *System Biology*

Submitted for the MRes in Computational Methods in Ecology and Evolution

Declaration

Dr. Martin supported me the original dataset while I was responsible for data cleaning and processing including manually segmented the fossil data. I developed a model called GIU-Net by myself. My supervisor provided me with an insight of fossil CT image processing and a guidance in thesis writing.

Version dated: August 29, 2019

Artificial intelligence approaches to optimising segmentation in computed tomography data processing

YUHENG WANG[1]

[1]*Department of Life Science, Imperial College London, Silwood Park, Ascot, SL5 7TG, United Kingdom*

Corresponding author: DR. MARTIN D. BRAZEAU, Department of Life Science, Imperial College London, Silwood Park, Ascot, SL5 7TG, United Kingdom; E-mail: cm.brazeau@imperial.ac.uk.

Abstract.—Paleontological research increasingly uses Computed tomography (CT) to study the architecture of fossils since CT techniques can extract three-dimensional representations of fossil structures without a destructive process (Dierick et al. 2007). After obtaining the CT scan, the following quantitative and qualitative analysis are both based on precise segmentation while not too much attention has been paid to the process of segmentation in the paleontological study. Traditional fossil segmentation methods are either time consuming or not precise and accurate enough. Previously, U-Net (Ronneberger et al. 2015) based approaches were proposed to segment medical CT imaging data.

However, the plain U-Net structure can not meet the requirement of fossil segmentation due to lack of global scene and smooth information. On the other hand, post-processing structure could provide large-scale global information but at the expense of time and complexity (Chen et al. 2016). Here we present the Gaussian Inception U-Net (referred to as GIU-Net) to capture the global smooth information while maintaining a simple structure of U-Net. The GIU-Net contains an inception-like (Szegedy et al. 2015) block to force U-Net to learn smooth and global information. We applied the proposed network to several fossil segmentation tasks and comprehensive results show that the GIU-Net outperforms the original U-Net method. We also proposed a probability averaging algorithm to better segment 3D data which is proven to be simple and effective.
(Keywords: Convolutional Neural Network, U-Net, Gaussian Pyramid, Semantic Segmentation, Fossil CT Imaging)

Computed tomography (CT) is now widely used in paleontology to study fossils. It can produce a 3D virtual model of fossils from a sequence of 2D x-ray images. It is a non-destructive method and needs little or no sample preparation (Dierick et al. 2007). With the great improvement of CT technology, the bottleneck of CT-based analyses transfers from scanning to data processing (Abel et al. 2012). The first step of data processing is segmentation, which means partitioning of an image into non-overlapping and constituent regions based on some characteristics such as intensity or texture (Pham et al. 2000). In other words, segmentation means separate one or several regions of interest (fossil in this case) out from original data (whole CT scan). This is also called semantic

segmentation in the computer version. In CT image processing, segmentation is extremely vital in both quantitative and qualitative analysis when the precise boundaries of the structure are needed. This is widely recognized in the medical industry. However, in the field of paleontology, not too much attention has been paid to the process of segmentation, resulting in the lack of convenient and consistence tissue/air and tissue/matrix segmentation methods (Scherf and Tilgner 2009).

Traditional fossil data segmentation includes four steps: contrast enhancement; surface determination; region growing and masking tools (Abel et al. 2012). During the procedure of segmentation, manual selections (and therefore subjectivity and error) are inevitable and time-consuming because of the diminishing image contrast at the junction of fossil and matrix. A few efforts have been made in fossil segmentation automation. Scherf and Tilgner (2009) proposed an algorithm that uses a 3D-Sobel filter to mark voxels at the peak of rapid changes in gray-scale values to detect the edge, thus segmenting the target. Dunmore et al. (2018) introduce a non-supervising machine-learning method named medical image analysis (MIA)-Clustering which is based on fuzzy c-means clustering (Pham and Prince 1999). However, these methods still require subjective hyper-parameters which are hard to determine. At the same time, when dealing with tough segmentation problems, their accuracy and robustness are far below the expected level.

Currently, neural network is the most popular deep learning method, where the convolutional neural network (CNN) is most commonly applied in image processing. CNN is designed to process data in the form of multiple arrays such as image, containing convolution layers and pooling layers as its unique structure (Lecun et al. 2015). The convolution layer is used for extracting features and the pooling layer is used for expanding the field of view. As a nonlinear transformation, CNN does not need handcrafted features to extract information from the image. Among CNN, the fully convolutional neural network (FCN) (Shelhamer et al. 2017) is specially designed for semantic segmentation

since it can accept image input of any size, while it can not give very fine and accurate segmentation because it does not have a sophisticated decoder structure.

There are two main ideas for solving this problem. One is to add post-processing structure to the neural network to refine the results, including conditional random fields (CRF), markov random fields (MRF), and other probabilistic graphical models (PGMs) (Chen et al. 2016; Kamnitsas et al. 2016; Shakeri et al. 2016). By using Gaussian filter in the feature space, this structure can increase the connection between image pixels and provide a global view, resulting in sharp boundaries and fine-grained segmentations with fewer noise points (Krähenbühl and Koltun 2012). Gaussian filtering is a process of weighted averaging of the entire image. The value of each pixel is obtained by weighted averaging of itself and other pixel values in the neighborhood and the weight is determined by the Gaussian distribution. Another way is to design an exquisite decoder structure for FCN, such as U-Net (Ronneberger et al. 2015), SegNet (Badrinarayanan et al. 2017), and RefineNet (Lin et al. 2016). Benefiting from its simple and compact structure, U-Net is most commonly used in CT image processing.

Fossil segmentation requires both a global view and a subtle view under the condition of lacking prior knowledge. It's hard to be satisfied with a simple encoder-decoder structure. On the other hand, the neural network with a post-processing structure is too complicated and time-consuming. We found that the Gaussian pyramid (Adelson et al. 1984), which is a traditional image segmentation method, could help with fossil segmentation. Performing different degrees of Gaussian filtering on the images, these images constitute octave, the octave downsampling set constitutes a Gaussian pyramid. Gaussian pyramids can combine multi-scale information due to multiple filtering. Scale-invariant feature transform (SIFT) algorithm (Lowe 1999) uses Gaussian pyramid to find features that are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. These features can be used

to optimize the fossil segmentation.

Zheng et al. (2015) and Liu et al. (2015) used convolution layers to approximate the CRF structure. Inspired by them, we added several convolution layers mimicking Gaussian filters at the beginning of the U-Net to help produce the approximation of Gaussian pyramid. With the instruction of Inception network (Szegedy et al. 2015), we then organized these structures into an inception module followed by two convolution layers. The whole module was added to the lowest skip connection of U-Net. We called the new structure Gaussian Inception U-Net (GIU-Net). In our experiments, we firstly applied a slightly modified U-Net structure to segment 2D fossil CT sequence and achieved state-of-art results. It was then used as our baseline model. The proposed model achieves superiority over U-Net performance by up to 1.6% in the Jaccard index and 0.5% in the Dice coefficient. It also can be visualized that the segmentation from the proposed model tends to have sharper and smoother boundaries than the prediction from U-Net. Since our task is the 3D binary segmentation, we adapted the majority voting (Zhou et al. 2016) method to a three direction probability averaging algorithm to segment voxel from 3D CT project. This method does not have any parameters to learn, and the time and memory requirements are diminished. Our main contributions are: (1) manually segmented a fossil fragment and used it as the ground truth; (2) applied U-Net to fossil segmentation, making it as a baseline; (3) proposed a new GIU-Net structure based on U-Net; (4) applied a new probability averaging algorithm to better segment 3D data.

Related Work

A large variety of work have been proposed to make improvement on basic U-Net. Çiçek et al. (2016) and Milletari et al. (2016) changed 2D convolution layers into 3D to better segment volumetric images. Oktay et al. (2018); Valloli and Mehta (2019); Li et al. (2019)

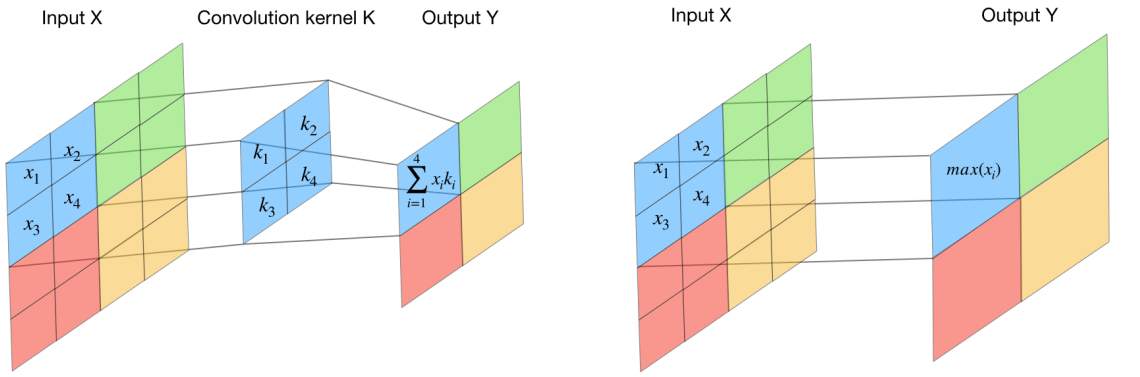
applied the attention gate into the skip connection to suppress irrelevant regions in an input image while highlighting salient features useful for a specific task. Squeeze & excitation (SE) block (Hu et al. 2017), as another format of attention unit, was introduced to U-Net by Roy et al. (2018) and Zhu et al. (2018). Jin et al. (2018) integrated the deformable convolution (Dai et al. 2017) into the proposed network to capture targets at various shapes and scales. Residual module (He et al. 2015a) can address vanishing gradient problem to some extent by summing up a shortcut connection with the residual function. It has been used in U-Net to substitute convolutional module (Alom et al. 2018; Ibtehaz and Rahman 2019) or even skip connection (Ibtehaz and Rahman 2019). Inception module (Szegedy et al. 2015) is essential in building a wider network that has a better multi-scale vision. It was achieved by concatenating a series of feature maps produced by different size convolution kernels. Theoretically, large size filter has a similar function as multiple small size filters (Szegedy et al. 2015), It has been proven to be practical in U-Net by Ibtehaz and Rahman (2019) and Gu et al. (2019). Zhang et al. (2018) and Li et al. (2017) used the dense connection (Huang et al. 2016) to reuse features thus reducing parameters. Zhou et al. (2018) also applied dense module to improve skip connection in U-Net. All these efforts have been made to improve networks' receptive ability and fitting ability, while pixels are still relatively isolated without PGMs' smoother term. Additionally, although deeper networks like ResNet (He et al. 2015a) have a very large receptive field, studies show that the network tends to gather information from a much smaller region (valid receptive filed) (Peng et al. 2017). Some networks used dense connection or inception module to alleviate this drawback, but their kernel size is still too small (up to 7*7) to get the global scene. Unlike medical image segmentation, fossil data does not have any prior in shape or color, which makes the global scene extremely important.

There is also a lot of work dedicated to integrating the PGM structures into neural networks. Zheng et al. (2015) combined CRF into CNN as a recurrent neural network

(RNN) structure, making it an end-to-end algorithm. But the CRF structure in their algorithm takes 5 to 10 iterations to converge, which is time-consuming. Liu et al. (2015) managed to use only one iteration to infer the MRF. However, during the training phase, they have to infer the parameters separately. In detail, they firstly train the CNN without MRF structure, then learning MRF while fixing the parameters of CNN, which makes the learning step complicate and hard to repeat. Different from the previous methods, the proposed method skillfully integrates Gaussian pyramid and U-Net structure, so that it has both the smoothing effect of the post-processing structure and the simple structure of U-Net. The proposed GIU-Net is also an end-to-end structure which is of benefit to practical application.

Method

Our improvements are based on U-Net's encoder-decoder structure. It consists of a series of connected basic components including convolution, pooling, and activation functions. The operation of the inner product (multiple-by-element and summation) on the image and the filter matrix (also named kernel) is a so-called convolution operation (see Fig. 1(a)). The convolution filter could either be 2D or 3D, depending on the number of channels of the input feature map. The convolution filter moves at a certain step size (stride) on the input feature map to generate an output. Multiple filters overlay contribute to convolution layers, and the number of filters determines the output channels number. The convolution operation is a linear transformation, and if without activation function, no matter how many layers of the neural network, the output is a linear combination of inputs. This is equivalent to the effect of no hidden layer, then the network's approximation ability is quite limited. When introducing non-linear function as activation function, the neural network can almost approximate arbitrary functions. We used the rectified linear unit



(a) The 2×2 convolution operation

(b) The 2×2 max pooling operation

Figure 1: Convolution and max-pooling operation. The convolution means multiple-by-element and summation operation of the input feature map and the convolution kernel. In the diagram, the size of the convolution kernel is 2×2. The max-pooling means down-sampling with a maximum value in a certain size area, which is 2×2 in this case.

(ReLU) (Lecun et al. 2015) and Sigmoid function in our neural network (see Fig. 2). In ReLU function we have $f(x) = max(x, 0)$, meaning that neurons with negative values will be suppressed while positive neurons will not be affected. This simulates the sparse activation of biological neurons (Attwell and Laughlin 2001) and being easy to calculate. The expression of the Sigmoid function is $f(x) = \frac{1}{1+e^{-x}}$, Sigmoid can map the convolutional layer output to $(0, 1)$, thus outputting probabilistic predictions. In pooling operation, the input image is divided into many rectangular areas (can overlap), followed by the output of the average, minimum, or maximum value for each sub-area, corresponding to mean-pooling, min-pooling, and max-pooling (see Fig. 1(b)) respectively. The interval between adjacent rectangular areas is called the stride. Pooling, a down-sampling operation, is equivalent to making dimensional reductions in the spatial (width, height) range, allowing the model to extract a wider range of features. At the same time, the input size of the next layer is reduced, thereby reducing the amount of calculation and the number of parameters. Information is passed through the convolution and pooling layers in

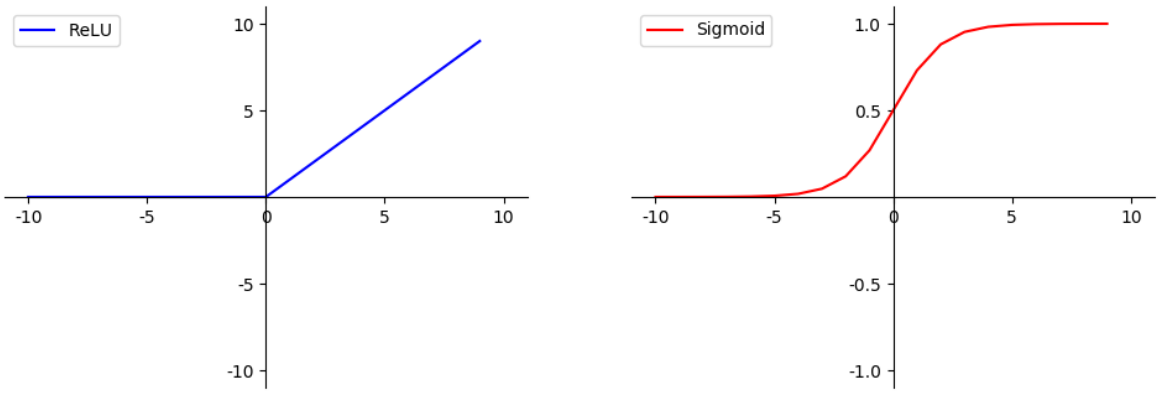


Figure 2: ReLU and Sigmoid activation function. RelU can suppress the negative value in the output feature map of convolution layers and Sigmoid can map the convolution output into probability.

the convolution neural network, allowing acquisition of latent feature in a very high dimension.

As shown in the figure 3, the basic U-Net structure consists of a contracting path as an encoder and a symmetrical expanding path as a decoder. The encoder is used to capture context and extract high dimension features, while decoder could decrease the dimension of feature maps to restore the pixel-wise segmentation prediction that has the same size and dimension of the input image. The skip connection directly copies and concatenate the feature map from the corresponding encoder to the decoder, combining high-resolution features that have precise localization information with the up-sampled output. These features of the encoder-decoder network give the neural network the ability to accept almost any size input and output the same size accurate pixel-wise predictions. It is noticeable that U-Net structure involves reducing the width and height of the picture to 1/32, so the width and height of the input image are required to be an integral multiple of 32.

The encoder consists of the several application of two 3×3 same convolutions,

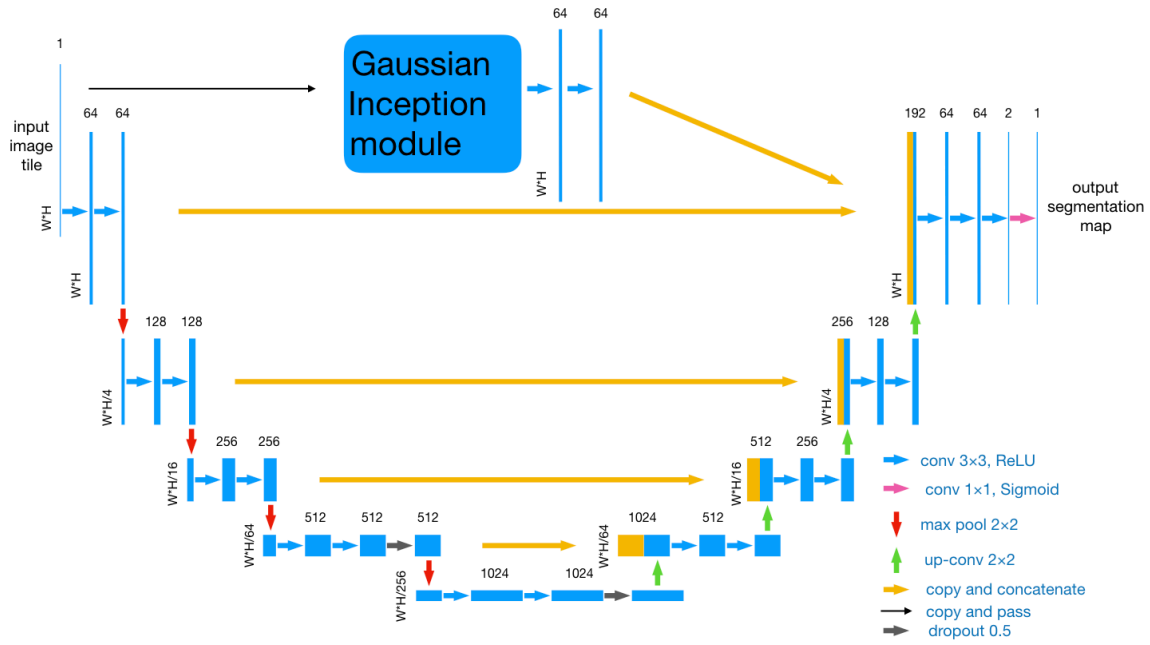
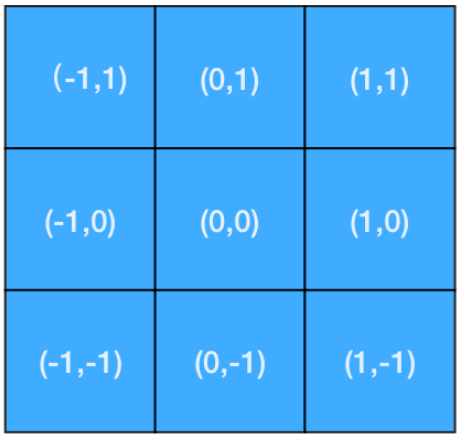


Figure 3: The structure of GIU-Net. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The spatial size is provided at the lower-left edge of the box. Yellow boxes represent copied feature maps from skip connection. The arrows denote the different operations. In GIU-Net, the GIU module alongside with two convolution layers are added to the basic U-Net structure.

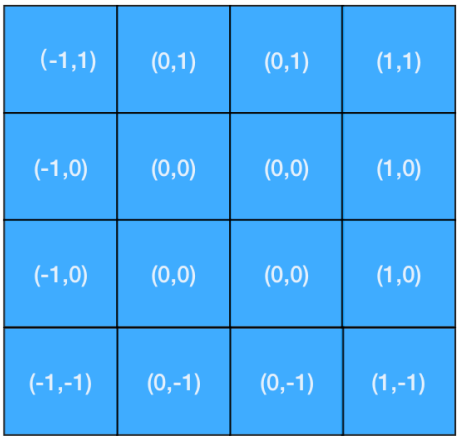
followed by a ReLU and a 2×2 max pooling operation with stride 2 for down-sampling. Same means padding the input feature map with zero to output the same spatial size feature maps. Except for the first convolution which changes the number of image channels from 1 to 64, the convolution layers in the encoder gradually double the channels into 1024. At the same time, four max-pooling layers reduce the image spatial size to 1/256 correspondingly. Apart from these structures, we added two dropout (Srivastava et al. 2014) layers with 50% probability at the fourth and fifth steps of the encoder to alleviate over-fitting. Dropout refers to the temporary discarding of neural network units from the network according to a certain probability during the training of the deep learning

network. According to the author's point of view, it forces a neural unit to work with other randomly selected neural units to achieve good results, reducing the joint adaptability between adjacent neuron nodes and enhances generalization ability (Srivastava et al. 2014). Every step in the decoder begins with an up-convolution, consisting of an up-sampling and a 2×2 convolution which halves the number of channels. Then the feature map is concatenated with corresponding one from the encoder, followed by two 3×3 convolution layer with the ReLU as their activation function. At last, a 1×1 convolution with Sigmoid activation function is used to output one prediction map containing the probability of target foreground.

In our proposed GIU-Net, we added GIU module and two convolution layers into the U-Net (see Fig. 3). The GIU module can generate an approximating Gaussian pyramid (Adelson et al. 1984) and it consists of many parallel convolution layers mimicking the Gaussian filters. This is inspired by Zheng et al. (2015) and Liu et al. (2015) who used specially designed convolution layer to approximate the Gaussian filtering process in CRF and MRF. Since the nature of convolution is filtering, we realize this simply by initializing the convolution kernel with the parameters of the 2D Gaussian filter and making it untrainable, which means these parameters can not be updated when training the neural network. The Gaussian filter we used here is discrete because the process of convolution is discrete. Discrete Gaussian filter is determined by Gaussian distribution, where standard derivation σ determines the magnitude of the distribution. For a $(2k+1)(2k+1)$ size discrete Gaussian filter, its parameters are $G_{i,j} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}}$. Two-dimensional coordinates i, j can be obtained by establishing a Cartesian coordinate system from the Gaussian filter center (see Fig. 4(a)). It is worth noting that the Gaussian filter has only an odd size. However, we also used some slightly modified Gaussian filter which is even size (see Fig. 4(b)). Unlike the strict Gaussian filter, the four central pixels of even size filter have the same weight. This design gave better results in our pre-test.



(a) Discrete Gaussian filter coordinates example



(b) Even size filter coordinates example

Figure 4: The example of discrete Gaussian filter and even size filter coordinates. Even size filter is slightly modified from Gaussian filter by replicating the center of it.

Through experiment I, we set the number of convolution layers in GIU module to 4. Learning from the way of SIFT (Lowe 2004) building the first octave of Gaussian pyramid, we set the first kernel with standard deviation σ_0 as 1.52, the following kernel has their standard deviation with the law that $\sigma_s = \sigma_0 2^{s/S}$, which are 1.92, 2.41, and 3.04. The larger the standard deviation σ_0, the larger the range of influence of the filter, meaning that the pixel is affected by farther pixels. s is the serial number of the kernel. S affects the number of layers in an octave, and we set it as 3, which is a widely used one in Gaussian pyramid. In practical applications, when calculating the discrete approximation of the Gaussian function, pixels outside the approximate 3σ distance can be regarded as ineffective, and the calculation of these pixels can be ignored, defined as a truncated Gaussian. Usually, the image processing program only needs to calculate the matrix of $(6\sigma + 1) * (6\sigma + 1)$ to ensure the influence of related pixels. So we set the size of convolution kernels as 11, 13, 16, and 20. These convolution layers use same padding to ensure the same size output. The original image forms an approximating octave after passing these certain fixed convolution layers. We then concatenate them together to form an Inception-like (Szegedy et al. 2015)

block (see Fig. 5). This structure can also be seen as a pyramid module in the view of

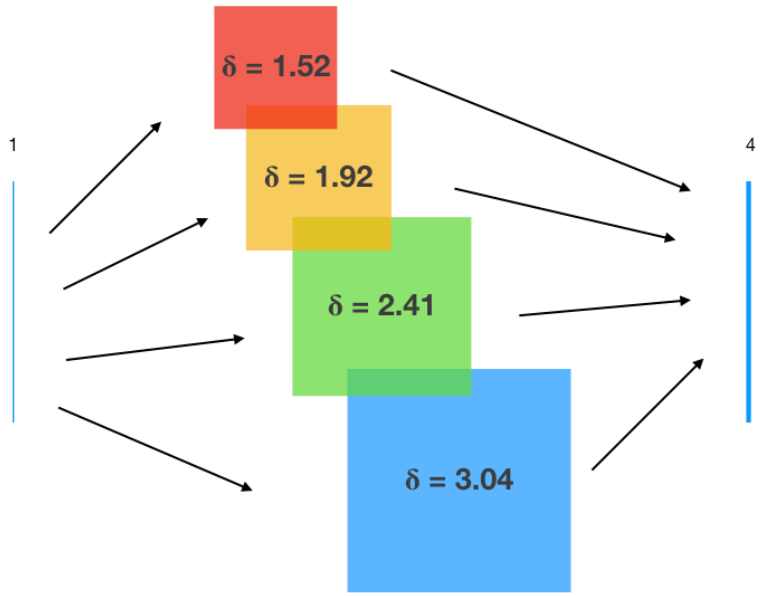


Figure 5: The structure of GIU module. The four parallel convolution kernels are used to approximate the Gaussian filters, which have standard deviation of 1.52, 1.92, 2.41, and 3.04. The original image forms four corresponding filtered feature map after passing these convolution kernels. We then concatenated them together to form an Inception-like module.

Pyramid Scene Parsing Network (Zhao et al. 2016). The following two normal convolution layers were used to extract scale-invariant and global scene features from the octave. These features were directly concatenated to the last up-convolution layer alongside with skip-connection from the first block in the encoder, as shown in the figure. 3. It has not been down-sampled and up-sampled by the neural network, retaining high-resolution scale-invariant smoothen information, which can better segment the fossil data.

3D-to-2D Image Sampling and 2D-to-3D Probability Averaging

We used a 3D-to-2D image sampling and 2D-to-3D probability averaging pipeline to segment 3D data (see Fig. 6). In the proposed approach, we firstly decomposed the 3D

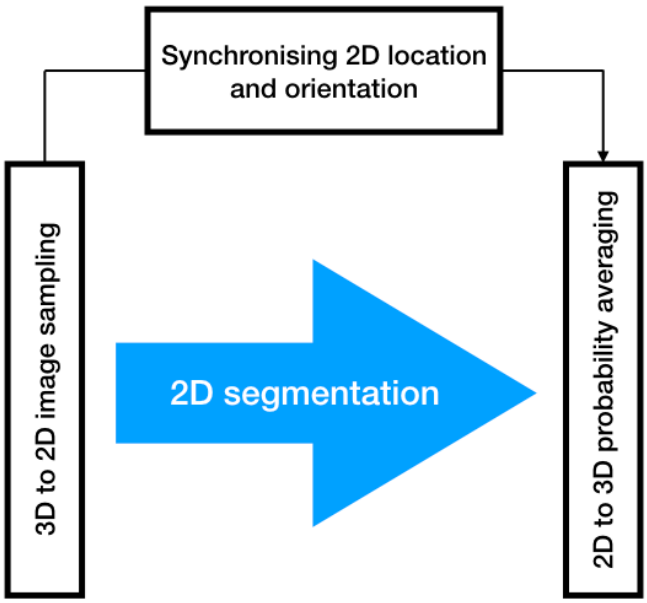


Figure 6: The pipeline used to segment 3D data. In this method, we decomposed 3D data into three orientation 2D image sequences. Then we segmented 2D image sequences with 2D segmentation methods, followed by reconstructing 3D segmentation prediction with three orientation probability predictions.

data into 2D image sequences with three directions: axial, coronal, and sagittal. Then we segmented all 2D images with 2D segmentation methods, which are U-Net and GIU-net. We then stack 2D image sequences back to 3D data, getting triple 3D probability maps from axial, coronal, and sagittal respectively, and each voxel is annotated three times from different directions. After synchronizing the location and orientation of three predictions, we averaged the probability to get the final segmentation map. Benefiting from redundantly labeling the voxel with different orientations, this method could increase the robustness and accuracy of segmentation. When compared with other 3D segmentation methods like 3D U-Net (Çiçek et al. 2016) and V-Net (Milletari et al. 2016), this method have no parameters to learn and hardly takes up memory and time resources.

Training and Evaluation

Neural network has the process of forward propagation and backward propagation in the training step. During forward propagation, the proposed network can pass information of the input image layer by layer to get the point-by-point segmentation prediction. Then predictive segmentation is compared to the ground truth and the loss function is used to calculate the error. In the backward propagation, errors are passed back and forth in the neural network by the chain rule (Rumelhart et al. 1988). The parameters of the convolution layers are also updated by the optimizer from the back to the front simultaneously. One forward propagation and one back propagation form an iteration, training the neural network with one batch of data. Through repeated iterations, the proposed network should give accurate segmentation predictions. We could use the evaluation matrix to test the performance of the neural network.

During the training step, we initialize the normal convolution layers with HE initialization to ensure that input and output can maintain a normal distribution with similar variance (He et al. 2015b), which is important for neural network to converge. Define y_i as the ground truth in a binary segmentation, and $y_i \in \{0, 1\}$. $\hat{y}_i$ denotes the prediction of each point whose value is a probability between 0 and 1. n represents the number of pixels or voxels in one batch data. We used binary cross-entropy loss (see equation 1) as loss function since our networks output a probability of binomial.

$$Loss = -\frac{1}{n}\sum_{i}^{n}(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)) \tag{1}$$

Compared to the traditional MSE loss, cross-entropy can make the neural network learn faster because it is an exponential loss. We adapted ADAM (Kingma and Ba 2014) as the optimizer update the parameters in GIU-Net because it is widely used in computer vision processing benefiting from its efficient and concise calculations. Learning rate is the update

ratio of weights for the optimizer, a larger learning rate means that weights are updated faster, and vice versa. If the learning rate is too small, the neural network will learn too slowly. If it is too large, the neural network will fall into the local optimal solution, and even the loss will become bigger. Through experiments, we chose 2e-4 as the learning rate for U-Net, and 2.5e-4 for GIU-Net. We trained the networks with batch size equalling to 2, which means two pictures are sent to the neural network and optimized simultaneously. We set the epoch numbers as 100, meaning that the neural network traverses all training data 100 times. We trained the neural network using TensorFlow (Abadi et al. 2015) backend Keras (Chollet et al. 2015) package with GPU (Tesla K80) based on Google Colab. All programs were run on python 3.5.

Define p_i as the binary prediction of the $\hat{y}_i$ which binarizes $\hat{y}_i$ to 0 or 1 according to the threshold. We first evaluated segmentation performance using Dice similarity coefficient (DSC) since it is the most used metric in validating medical volume segmentations (Taha and Hanbury 2015). DSC calculates the degree of similarity between predictions and the ground truth based on their overlap. The larger the value of DSC, the closer the segmentation prediction is to the ground truth. It is defined by equation 2,

$$DSC = \frac{2\sum_i^n y_i p_i}{\sum_i^n y_i^2 + \sum_i^n {p_i}^2} \tag{2}$$

and we used 0.5 as our threshold, meaning that all values greater than 0.5 are mapped to 1 and all values less than or equal to 0.5 are mapped to 0. During Training, we also used DSC to select best model with the help of validation set. Jacard index, also known as intersection over union (IOU) was also used in our experiment (see equation 3).

$$IOU = \frac{\sum_i^n y_i p_i}{\sum_i^n y_i^2 + \sum_i^n {p_i}^2 - \sum_i^n y_i p_i} \tag{3}$$

It is very similar to DSC coefficient but we use multiple threshold averaging instead of single threshold to test the robustness of segmentation. We used thresholds from 0.5 to 9.5, every 0.05 as a interval.

Data

The sample is a piece of bedrock from the Early Devonian Tsagaan Salaa Formation of western Mongolia. It preserves dermal bone fragments of placoderm fishes. The bulk of the sample consists of siliciclastic rock: rocks primarily made of silicate minerals derived from weathering and deposition of nearby source rocks. The bone fragments consist either of hydroxyapatite minerals which are mainly formed of calcium phosphate. These two constituents provide good contrast in x-ray tomography. However, the bulkiness of the sample means that CT results will be somewhat imperfect and require some manual segmentation in order to process. Therefore, this sample makes for a good initial target for the development of AI techniques for working on fossils. The CT scanning produced by 210kV, 200μA X-ray with 1.500mm Copper filter. The voxel size is 0.0624 in both three directions. The raw data is a 16-bit grayscale image in TIFF format, containing 1479×1283×1403 voxels.

Implementation

The raw data consists of three fossil fragments, and we named the fossils as A, B, and C in ascending order of segmentation difficulty. Under the guidance of the supervisor, I manually segmented fossil A with Mimics Research 19.0 for Windows, making it into a ground truth for neural network training and testing. Then we used a trained neural network to predict fossil B and C to see if the proposed network can successfully segment more complicated data or not. We manually cropped the raw data as well as segmented

data for fossil A with ImageJ-win64, making both of them into 355×381×261 voxels, followed by re-sampling them into three directions: axial, coronal, and sagittal. We then saved these data into three separate folders where we can read the image from.

In experiment I, we read all image and segmented files (also called mask files) from separate folders and resize all of the pictures into 256×256. This is not necessary but in order to improve training efficiency and save memory space. Also, we divided them by 65535, mapping the value between 0 and 1, which is helpful for network probabilistic segmentation because neural network can converge better when the input and output are both between 0 and 1. We then separated the corresponding image and mask data into training set, validation set, and test set by a ratio of 3:1:1. We used training set to train the network, validation set to help set hyper-parameters including saving the best neural network parameters during training, and test set to verify the performance of neural network. U-Net was used as a state-of-art baseline for comparison. GIU-Net with different numbers of convolution layers in GIU module were tested and the best was chosen for further test. For fossil B and C, we manually cropped their raw data into 282×212×224 and 884×569×403 voxels, followed by resizing them into 256×256×256 and 480×352×256 voxels respectively due to the calculating memory limits. We then used 3D-to-2D image sampling and 2D-to-3D probability averaging pipeline to predict their segmentation map. Since there is no manual segmentation ground truth for B and C, we did not compare the performance of U-Net and GIU-Net in this step.

In experiment II, instead of reading all three orientation image sequences and doing cross-validation, we take turns to train with two of the three orientations data and used the other for validation and testing. For example, we trained the network with the combining of axial and coronal data and tested on sagittal data. These could increase the robustness of the results while avoiding data leakage since data from different orientations tend to have larger diverge in distribution. We used experiment results to compare the performance of

U-Net and GIU-Net in 2D segmentation, then combining three direction prediction using probability averaging algorithm to compare them in 3D segmentation. We also calculated the individual prediction performance for each direction to prove that the probability averaging algorithm is effective. The training process for all neural networks was repeated 5 times and the neural network whose parameters obtained the best results on the validation set were taken. This is to ensure the reliability of the experimental results.

Results

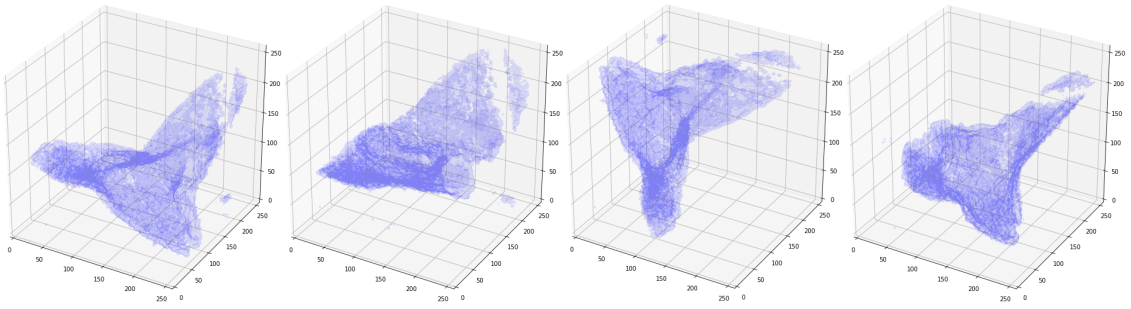
Through experiment I, we compared the performance of GIU-Net with different number of convolution layers in GIU module as well as plain U-Net structure in 2D segmentation (see Table 1). The number of convolution layers are from 1 to 6, corresponding to gaussian filters with standard deviation as 1.52, 1.92, 2.41, 3.04, 3.83, and 4.83. It is shown that the

Method		Dice similarity coefficient (DSC)	Intersection over union (IOU)	Cross-entropy loss
GIU-Net	Number of convolution layers			
	1	0.958	0.872	0.00607
	2	0.959	0.873	0.00595
	3	0.958	0.877	0.00595
	4	**0.961**	**0.888**	**0.00587**
	5	0.959	0.875	0.00592
	6	0.960	0.875	0.00623
U-Net		0.957	0.877	0.00832

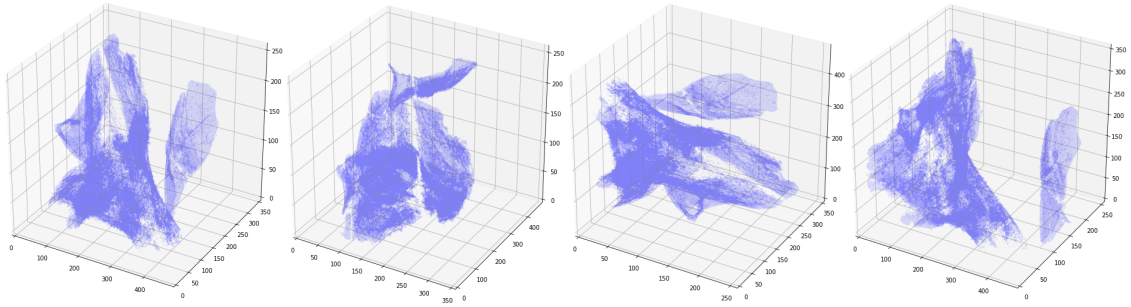
Table 1: The performance of U-Net and GIU-Net with different number of convolution layers in 2D segmentation. The GIU-Net with four convolution layers in GIU module gets the best performance on DSC, IOU, and cross-entropy loss.

GIU-Net with four convolution layers gets the best performance on both DSC, IOU, and cross-entropy loss. The plain U-Net structure already gets a state-of-art performance with DSC value at 0.957, IOU value at 0.877, and cross-entropy loss at 0.00832. The proposed GIU-Net outperforms the U-Net by 0.4% on DSC, meaning that it can better segment the

fossil. It also has better robustness since it beats U-Net by 1.1% on IOU with multiple thresholds. It is worth noting that the cross-entropy loss of GIU-Net is much lower than that of U-Net in all possible numbers of convolution layer. We also got the 3D segmentation prediction of fossil B and C from GIU-Net and U-Net and they were both well segmented. Because we did not compare the performance of two network structure, we only demonstrate the segmentation results from GIU-Net here (see Fig. 7). The predict from U-Net is put on the supplementary information. (see Fig. 9) The segmentation results



(a) Segmentation results for fossil B showing in four directions



(b) Segmentation results for fossil C showing in four directions

Figure 7: The 3D segmentation results for fossil B and C showing in four directions. It can be seen that the edge of the segmentation result is sharp and clear with only a few noise points.

are good enough for further analysis in paleontology.

Through experiment II, we got the performance from three orientation test data. We only used GIU-Net with four convolution layers since it outperformed others in the

previous test. Table 2 shows that the GIU-Net outperforms U-Net in three orientation 2D

Method	Evaluation standard	Orientation of test data		
		Axial	Coronal	Sagittal
U-Net	DSC	0.940	0.941	0.945
	IOU	0.766	0.810	0.805
	Cross-entropy loss	0.0128	0.0162	0.0098
GIU-Net	DSC	**0.942**	**0.946**	**0.946**
	IOU	**0.772**	**0.826**	**0.814**
	Cross-entropy loss	**0.0095**	**0.0085**	**0.0079**

Table 2: The performance of GIU-Net and U-Net on three orientation 2D test image. It is shown that the GIU-Net beats classic U-Net in all test directions.

segmentation tests. From 3D segmentation test (see Table 3), we can see that GIU-Net

Method	Dice similarity coefficient (DSC)			
	Axial prediction	Coronal prediction	Sagittal prediction	Probability averaging prediction
U-Net	**0.9517**	0.9433	0.9436	0.9515
GIU-Net	0.9526	0.9486	0.9459	**0.9531**

Table 3: The performance of U-Net and GIU-Net in 3D segmentation with three individual direction prediction and probability averaging prediction. The table shows that the GIU-Net is better than U-Net in 3D segmentation test. And In most cases, the probability averaging prediction is better than the unidirectional prediction.

with probability averaging algorithm got the best performance. The probability averaging algorithm was proven to be effective although U-Net got a better performance in axial prediction than in probability averaging prediction. We can also get an intuitive judgment from the 3D segmentation results producing by GIU-Net and U-Net with probability averaging algorithm (see Fig. 8). GIU-Net gives better segmentation results.

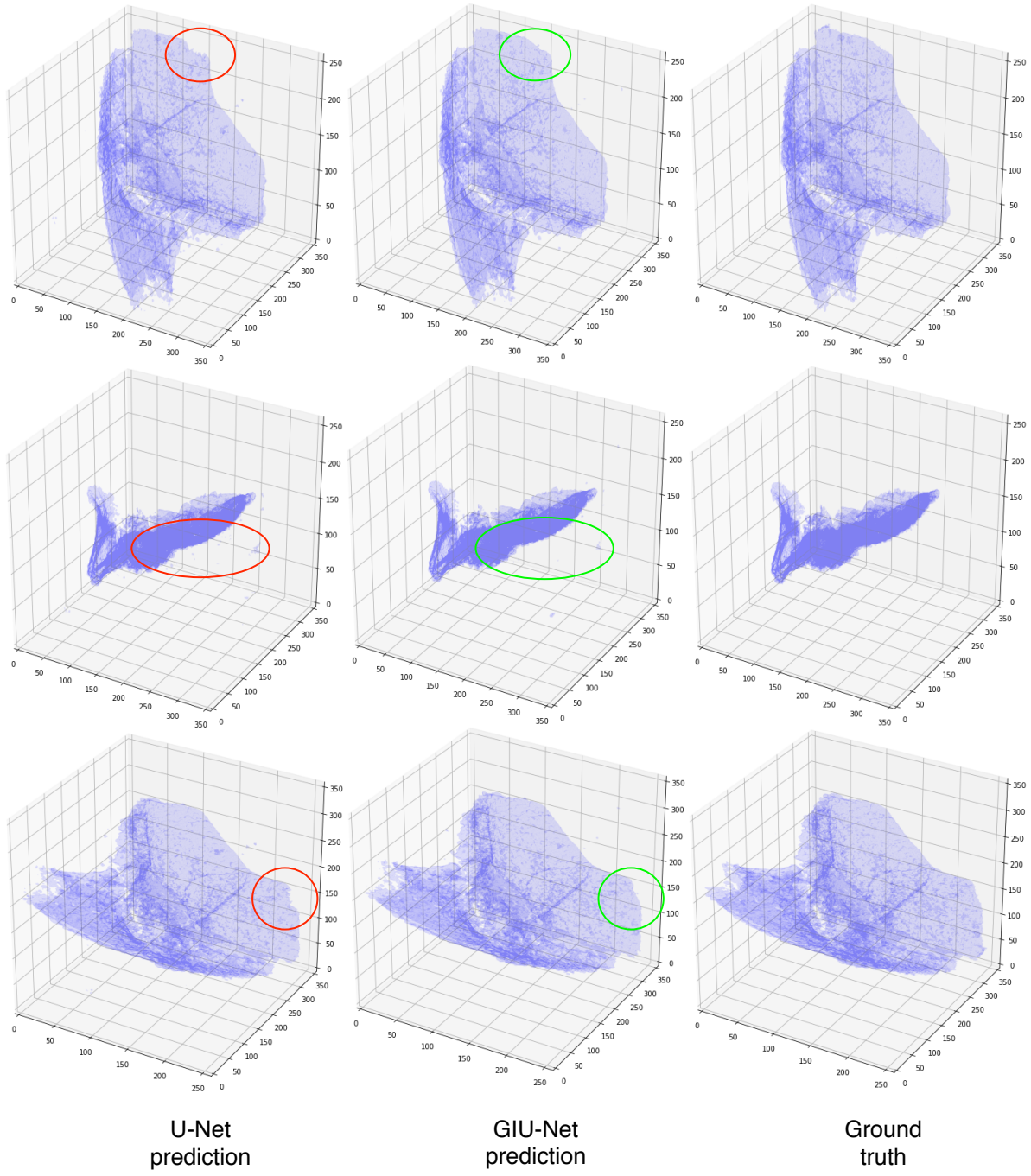


Figure 8: The segmentation results for fossil A from U-Net, GIU-Net, and ground truth. As we can see from the figure (especially inside the ellipse), the GIU-Net prediction has sharper and smoother edges and less noise, and is closer to the ground truth.

DISCUSSION

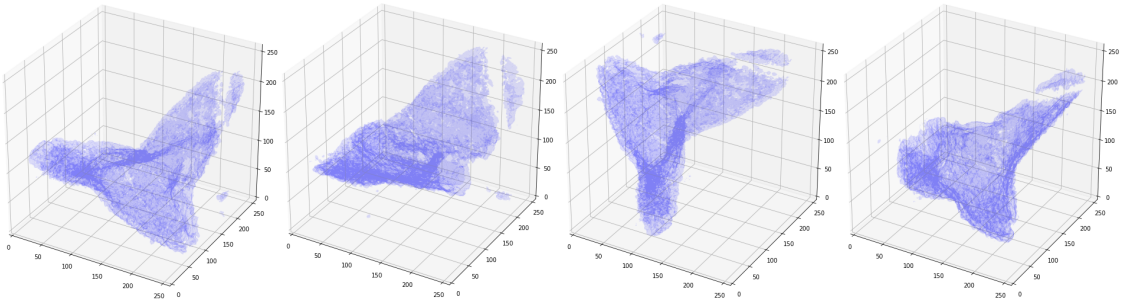
We presented GIU-Net, a skillful integration of U-Net and Gaussian pyramid. This structure has several benefits: (1) large size Gaussian kernel gives the network a better global scene which is greatly important to fossil data segmentation; (2) Gaussian filter acting as a smoothing term can sharpen the prediction boundaries and remove isolated noise to some extent; (3) two convolutional layers after Gaussian Inception (GI) module responsible for linear and non-linear transformation have the potential to find scale-invariant feature points in image, which also make a great contribution to edge segmentation; (4) this approach is based on an end-to-end learning without using any post-processing procedure. (5) just like U-Net, GIU-Net does not require a lot of data for training and can give good results. We also proposed a three direction probability averaging algorithm to help segment 3D data. Comparing to segmenting fossil CT from one direction, this method can use spatial redundancy information to make decisions, requiring small extra memory and time consumption. In the case of manual labeling fossil fragments A, we successfully segmented B, C automatically by the neural network.

This project also has a lot of room for exploration and improvements. We should have a better insight on why the number of convolution layers should be set to four and why the even size convolution kernel in GIU module could have good performance. Our proposed architecture performs only slightly better than U-Net in DSC and IOU, but has a significantly lower loss. This may suggest that we should change our loss function to Dice loss (Sudre et al. 2017) or IOU loss (Yu et al. 2016) since they directly optimize the DSC and IOU. During data processing, we did not convert the grayscale value to the Hounsfield unit (HU) value due to the lack of some parameters. The HU scale is a linear transformation of the original linear attenuation coefficient measurement, which is a more ubiquitous unit in CT image processing. Unlike grayscale value, the HU value of the same

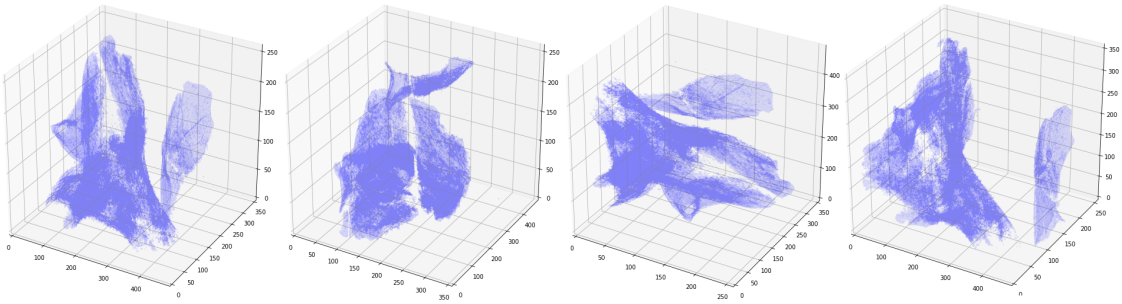
homogeneous medium is fixed, so we can remove some useless information by setting a interval to original images. Data argumentation like image rotation, re-scaling, or flipping can increase the stability of training and prevent over-fitting with the cost of time. It should be applied when better model generalization ability is pursued. In this project, we did not do any data normalization because the performance of the neural network without normalization was far better than that of adding one in our pre-test. The explanation may be all of our samples are from the same scan. In future applications, normalization should be considered when different fossil samples and different CT scanning parameters are used. Limited by computing resources, we did not apply GIU module in 3D U-Net structure, which is worthwhile trying in the future.

We confirmed that the convolutional neural networks can be efficiently applied to fossil segmentation. In the future, we will gradually test the practicality of neural networks with more complex tasks. Also, more medical segmentation (especially AI-based) methods should be introduced into paleontology applications.

Supplementary Information



(a) The prediction for fossil B



(b) The prediction for fossil C

Figure 9: The 3D segmentation results producing by U-net. It shows that U-Net can also produce good prediction despite some noise points

Acknowledgements

I am very grateful to my supervisor, Dr.Martin, for his time and his help. He has always been enthusiastic about this project and gave me several suggestions. He also supported me the dataset of the fossil CT scanning and gave me a good guidance in how to segment the fossil manually.

I also would like to thank my family for their help and their support during this whole year at Imperial College London. I am very grateful to them for giving me the opportunity to study in London.

Data and Code Availability

https://drive.google.com/drive/folders/1WQ-v_UfVRFvzRh8wk8BAR6wHuHJnJksp?usp=sharing

All the code and data are stored in Google drive. In order to facilitate the repetition, we did not upload the original CT file, but stored the processed image as NumPy array in .p files. It is worth noting that due to Googles access control, programs can not be run on original Google drive. You can either download the main directory (master_project) in the link and upload it to your own Google drive as a whole. Or contact me with yuheng.wang18@imperial.ac.uk, in which case I can share the username and password with anyone who need to repeat the program.

*

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia,

R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abel, R. L., C. R. Laurini, and M. Richter. 2012. A palaeobiologist ' s guide to virtual ' micro-CT preparation. Palaeontologica Electronica 15:496–500.

Adelson, E. H., C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. 1984. 1984, Pyramid methods in image processing. RCA Engineer 29:33–41.

Alom, M. Z., M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari. 2018. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. CoRR abs/1802.06955.

Attwell, D. and S. B. Laughlin. 2001. An energy budget for signaling in the grey matter of the brain. Journal of Cerebral Blood Flow & Metabolism 21:1133–1145 pMID: 11598490.

Badrinarayanan, V., A. Kendall, and R. Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39:2481–2495.

Chen, L., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. CoRR abs/1606.00915.

Chollet, F. et al. 2015. Keras. https://keras.io.

Çiçek, Ö., A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 2016. 3d u-net: Learning dense volumetric segmentation from sparse annotation. CoRR abs/1606.06650.

Dai, J., H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. 2017. Deformable convolutional networks. CoRR abs/1703.06211.

Dierick, M., V. Cnudde, B. Masschaele, J. Vlassenbroeck, L. Van Hoorebeke, and P. Jacobs. 2007. Micro-CT of fossils preserved in amber. Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 580:641–643.

Dunmore, C. J., G. Wollny, and M. M. Skinner. 2018. Mia-clustering: a novel method for segmentation of paleontological material. PeerJ 6:e4374.

Gu, Z., J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao, T. Zhang, S. Gao, and J. Liu. 2019. Ce-net: Context encoder network for 2d medical image segmentation. CoRR abs/1903.02740.

He, K., X. Zhang, S. Ren, and J. Sun. 2015a. Deep residual learning for image recognition. CoRR abs/1512.03385.

He, K., X. Zhang, S. Ren, and J. Sun. 2015b. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR abs/1502.01852.

Hu, J., L. Shen, and G. Sun. 2017. Squeeze-and-excitation networks. CoRR abs/1709.01507.

Huang, G., Z. Liu, and K. Q. Weinberger. 2016. Densely connected convolutional networks. CoRR abs/1608.06993.

Ibtehaz, N. and M. S. Rahman. 2019. Multiresunet : Rethinking the u-net architecture for multimodal biomedical image segmentation. CoRR abs/1902.04049.

Jin, Q., Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su. 2018. Dunet: A deformable network for retinal vessel segmentation. CoRR abs/1811.01206.

Kamnitsas, K., C. Ledig, V. F. J. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. 2016. Efficient multi-scale 3d CNN with fully connected CRF for accurate brain lesion segmentation. CoRR abs/1603.05959.

Kingma, D. P. and J. Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Krähenbühl, P. and V. Koltun. 2012. Efficient inference in fully connected crfs with gaussian edge potentials. CoRR abs/1210.5644.

Lecun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. Nature 521:436–444.

Li, R., M. Li, and J. Li. 2019. Connection sensitive attention U-NET for accurate retinal vessel segmentation. CoRR abs/1903.05558.

Li, X., H. Chen, X. Qi, Q. Dou, C. Fu, and P. Heng. 2017. H-denseunet: Hybrid densely connected unet for liver and liver tumor segmentation from CT volumes. CoRR abs/1709.07330.

Lin, G., A. Milan, C. Shen, and I. D. Reid. 2016. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. CoRR abs/1611.06612.

Liu, Z., X. Li, P. Luo, C. C. Loy, and X. Tang. 2015. Semantic image segmentation via deep parsing network. CoRR abs/1509.02634.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. Pages 1150– *in* Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2 ICCV '99 IEEE Computer Society, Washington, DC, USA.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60:91–110.

Milletari, F., N. Navab, and S. Ahmadi. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. CoRR abs/1606.04797.

Oktay, O., J. Schlemper, L. L. Folgoc, M. C. H. Lee, M. P. Heinrich, K. Misawa, K. Mori, S. G. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert. 2018. Attention u-net: Learning where to look for the pancreas. CoRR abs/1804.03999.

Peng, C., X. Zhang, G. Yu, G. Luo, and J. Sun. 2017. Large kernel matters - improve semantic segmentation by global convolutional network. CoRR abs/1703.02719.

Pham, D. and J. Prince. 1999. An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. Pattern Recognition Letters 20:57–68.

Pham, D. L., C. Xu, and J. L. Prince. 2000. Current methods in medical image segmentation. Annual Review of Biomedical Engineering 2:315–337 pMID: 11701515.

Ronneberger, O., P. Fischer, and T. Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. CoRR abs/1505.04597.

Roy, A. G., N. Navab, and C. Wachinger. 2018. Concurrent spatial and channel squeeze & excitation in fully convolutional networks. CoRR abs/1803.02579.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1988. Neurocomputing: Foundations of research. chap. Learning Representations by Back-propagating Errors, Pages 696–699. MIT Press, Cambridge, MA, USA.

Scherf, H. and R. Tilgner. 2009. A new high-resolution computed tomography (ct) segmentation method for trabecular bone architectural analysis. american journal of physical anthropology, 140, 39-51. American journal of physical anthropology 140:39–51.

Shakeri, M., S. Tsogkas, E. Ferrante, S. Lippé, S. Kadoury, N. Paragios, and I. Kokkinos. 2016. Sub-cortical brain structure segmentation using f-cnn's. CoRR abs/1602.02130.

Shelhamer, E., J. Long, and T. Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39:640–651.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15:1929–1958.

Sudre, C. H., W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso. 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. CoRR abs/1707.03237.

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2015. Rethinking the inception architecture for computer vision. CoRR abs/1512.00567.

Taha, A. A. and A. Hanbury. 2015. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. BMC Medical Imaging 15:29.

Valloli, V. K. and K. Mehta. 2019. W-net: Reinforced u-net for density map estimation. CoRR abs/1903.11249.

Yu, J., Y. Jiang, Z. Wang, Z. Cao, and T. S. Huang. 2016. Unitbox: An advanced object detection network. CoRR abs/1608.01471.

Zhang, J., Y. Jin, J. Xu, X. Xu, and Y. Zhang. 2018. Mdu-net: Multi-scale densely connected u-net for biomedical image segmentation. CoRR abs/1812.00352.

Zhao, H., J. Shi, X. Qi, X. Wang, and J. Jia. 2016. Pyramid scene parsing network. CoRR abs/1612.01105.

Zheng, S., S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. 2015. Conditional random fields as recurrent neural networks. CoRR abs/1502.03240.

Zhou, X., T. Ito, R. Takayama, S. Wang, T. Hara, and H. Fujita. 2016. Three-dimensional ct image segmentation by combining 2d fully convolutional network with 3d majority voting. Pages 111–120 *in* Deep Learning and Data Labeling for Medical Applications (G. Carneiro, D. Mateus, L. Peter, A. Bradley, J. M. R. S. Tavares, V. Belagiannis, J. P. Papa, J. C. Nascimento, M. Loog, Z. Lu, J. S. Cardoso, and J. Cornebise, eds.) Springer International Publishing, Cham.

Zhou, Z., M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. 2018. Unet++: A nested u-net architecture for medical image segmentation. CoRR abs/1807.10165.

Zhu, W., Y. Huang, H. Tang, Z. Qian, N. Du, W. Fan, and X. Xie. 2018. Anatomynet: Deep 3d squeeze-and-excitation u-nets for fast and fully automated whole-volume anatomical segmentation. CoRR abs/1808.05238.