This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2995367, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.DOI

AutoSegNet: An Automated Neural Network for Image Segmentation

ZHIMIN XU[1], SI ZUO[1], EDMUND Y. LAM[2](Fellow, IEEE), BYOUNGHO LEE[3](Fellow, IEEE), AND NI CHEN[3]

[1]SharpSight Limited, Science Park West Avenue, Shatin, Hong Kong, China (e-mail: zmxu@sharpsight.hk, szuo@sharpsight.hk)
[2]Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China (e-mail: elam@eee.hku.hk)
[3]Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, Korea (e-mail: byoungho@snu.ac.kr, nichen@snu.ac.kr)

Zhimin Xu and Si Zuo are co-first authors. Corresponding author: Ni Chen (e-mail: nichen@snu.ac.kr).

This work was supported in part by the Brain Korea 21 Plus project 2019, the National Natural Science Foundation of China under Grant 61705241 and Natural Science Foundation of Shanghai under Grant 17ZR1433800.

ABSTRACT Neural Architecture Search (NAS) has drawn significant attention as a tool for automatically constructing deep neural networks. The generated neural networks are mainly applied for image classification, and natural language processing. However, there are increasing demands for image segmentation in various areas, such as medical image processing, satellite image object location, and autopilot technology. We propose a NAS method called Automated Segmentation Network (AutoSegNet), targeting industrial and medical image segmentation. The search architectures are constructed by stacking the downsampling layer, the bridge layer, and the upsampling layer, which are explored by a recurrent neural network. Compared with other related methods for image segmentation, the proposed method has a small search space but can explore most of the-state-of-the-art supervised image segmentation models. We perform verification on two datasets, and the results show that AutoSegNet achieves superior segmentation results with clear and continuous segmented edges, as well as better image details.

INDEX TERMS Neural Architecture Search, Image Segmentation, Deep Neural Network

I. INTRODUCTION

Neural Architecture Search (NAS) aims to search for the best neural network architecture, given the learning dataset. Currently, it has been successfully applied for image classification and language modeling [1–4]. NAS mainly consists of two parts: a controller for generating architecture parameters of the neural network and a validation neural network for validating the given architecture parameters by constructing, training, and testing the network. The optimization of controller and validation network is based on reinforcement learning [5]. The accuracy of the trained network will be fed back to the controller as a reward and guide the controller to optimize continuously. Such a process will repeat for fixed epochs or stop while a specific parameter reaches a particular value.

Efficient Neural Architecture Search (ENAS) [3], targeting on image classification, generates the best neural network architecture with a fixed structure. The fixed structure is a traditional convolutional network with pooling layers. The ENAS generates the optimized neural network in two ways: figuring out the best component of each layer and searching the best combination of a layer, and generating the architecture by stacking layers.

To improve efficiency and reduce the amount of calculation, we generate the best architecture differently. As AutoSegNet for image segmentation, we have a fixed encoder-decoder structure, which is considered one of the most classic network structures for image segmentation [6–8]. The structure includes three types of layers: the downsampling layer, the bridge layer, and the upsampling layer. The downsampling layer reduces the input size so that the network can learn from the more significant receptive field. The upsampling layer works oppositely. Based on the features from the downsampling layer, the upsampling layer reconstructs the input image. A bridge layer lies in the middle of the whole network and connects the downsampling layers and the upsampling layer. Each layer includes several cells, and the number of cells can vary. The components of the cells are the parameters that need to be searched.

Unlike ENAS, which includes five operations for searching, AutoSegNet has a smaller search space, which significantly improves efficiency. As we have a fixed encoder-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2995367, IEEE Access

decoder network structure with the downsampling layers reducing the input size, the pooling operation is removed from the search space. Instead, a new hybrid dilated convolution [9, 10] operation is added to the search space. The hybrid dilated convolution is a kind of dilated convolution without the gridding effect. It processes the input with several convolution rates at the same time. In our cast, a group of rates is set to 1, 2, and 3. By doing so, the network can learn the input features from previous layers in a different receptive field without resolution reduction as well as grill effect. Consequently, compared with other NAS methods, we have a rather small search space with only four operations:

- 3×3 depthwise-separable convolution
- 5×5 depthwise-separable convolution
- 3×3 hybrid dilated convolution with rate 1, 2, 3
- Identity

Besides, skip connection plays a significant role in image segmentation. However, instead of manually adding the skip connection empirically [11], the proposed method searches both intra-cell skip connections and inter-cell skip connections, resulting in a more automated search network.

Meanwhile, unlike image classification, the features map for image segmentation task should not be too small. Otherwise, some details of the input image are difficult to reconstruct. To avoid too small feature size due to a deep network, the input feature is reduced to a fixed size. To achieve this, hybrid dilated convolution [10] is added to combine with the downsampling layers. An example of a fixed network structure is shown in Fig. 1. The proposed method is tested on an industrial segmentation dataset as well as a medical segmentation dataset, and the results show quality segmentation with clear and continuous segmented edges and better details in the segmentation.

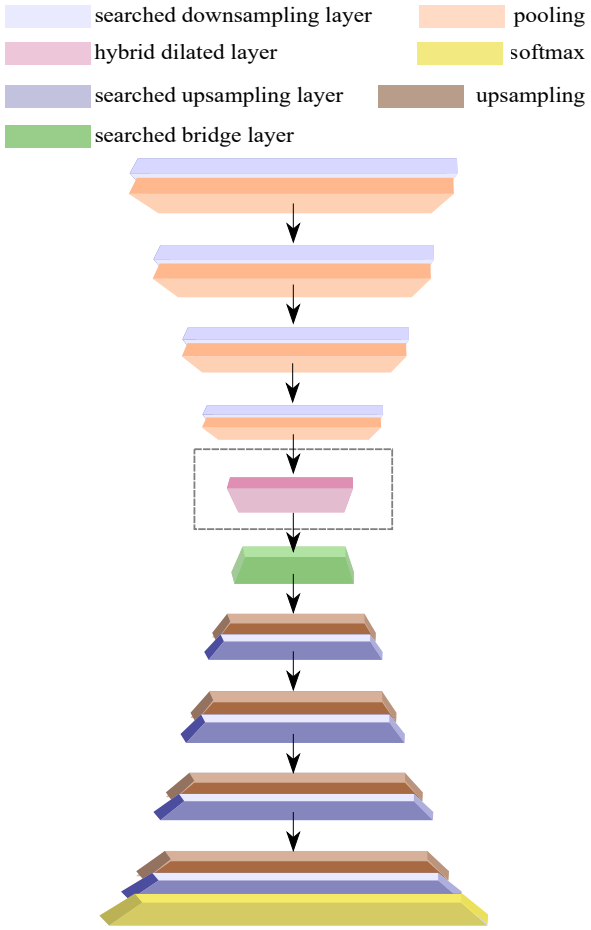


FIGURE 1: An example of a fixed network structure of the proposed method. The searched downsampling layer is followed by pooling layers. Upsampling is implemented before the searched upsampling layers. A searched bridge layer is added to connect the downsampling part and upsampling part and a softmax layer is added after the last searched upsampling layer for segmentation. The HDC layer will be added after the downsampling layers if the input of the block reaches the smallest acceptable size (such as 8×8) for preventing the resolution of the input being too small.

II. RELATED WORK

The proposed NAS method, as well as hyper-parameters optimization, are related to the previous work [1, 3]. The main challenges of neural architecture search can be divided into three parts: search space, searching strategies and evaluation methods.

The search space defines what operations will be searched and what kinds of networks can be generated. Theoretically, a larger search space covers more neural architectures. However, there is a trade-off between search space and efficiency - a vast search space results in longer searching time as well as more resource requirements. If the neural network is searched from scratch, for example, the height and width of kernel and stride and the number of filters, it would take around 800 GPU for 28 days to generate a convolutional neural network [1] given a CIFAR-10 [12] dataset. From the perspective of efficiency, some researchers limited the search space to a fixed number of operations. These operations are selected from the components of the-state-of-the-art models. Such a method significantly reduces the searching time and also achieves promising results on the same dataset [3, 13].

Meanwhile, the construction of the searched neural network architecture is different. At the beginning stage, the neural networks are searched layer by layer [1], which means each layer of the searched network might be different. However, from the perspective of efficiency, the entire network is constructed using a pre-defined pattern without the network level architecture search. Most of the pre-defined network patterns are simple. Taking NASNet as an example, only two types of cells needed to be searched: reduction cell, which downsizes the image resolution by two, and a normal cell, which keeps the resolution of the features. The final searched network is constructed by stacking the reduction cell and the normal cell. Targeting on image segmentation, the AutoDeepLab [14] proposed a network-level search space. In

AutoDeepLab, L layer is needed to be searched in a limited search space, which includes the downsampling layer, the upsampling layer, and the standard layer (for retaining the resolution of the features).

The searching strategy defines how to search through the search space and faces the exploration/exploitation trade-off during the searching stage. When choosing the next optimization step, on the one hand, a well-performing neural network architecture is expected to be found as quickly as possible. On the other hand, we should also avoid sinking in a suboptimal architecture. Some conventional search methods include random search, Bayesian optimization, evolutionary algorithms, reinforcement learning, and gradient-based algorithms. Among them, the reinforcement learning has become a competitive option since the proposal of NASNet [1]. NASNet achieved outstanding results on both CIFAR-10 and Penn Treebank benchmarks [15] with the search strategy based on reinforcement learning. In the NAS task, the generation of the architecture is treated as an agent. In the action selection phase, the reward is obtained through the prediction function on a test set and promotes the action selection. Almost all of the overall frameworks for problems related to NAS are based on such a strategy, but with different strategy representations and optimization algorithms.

An alternative way to use reinforcement learning is neural evolutionary that depends on evolutionary algorithms for the optimization of neural architecture. Such methods first randomly generate a population (with N sets of solutions). Then, they start to repeat the following steps: select, cross and mutate until the terminal condition is met. In Liu et al. [2], the generation starts from a small set of primitives such as convolutional and pooling operations at the bottom level of the hierarchy. Higher-level computation graphs, or motifs, are mutated from the lower-level motifs. Bayesian optimization is a common method for hyper-parameter optimization. In Jin et al. [16], Bayesian optimization is used with the help of the proposed neural network kernel and a tree-structured acquisition function optimization algorithm to accelerate the selection process of morph operation of neural architectures.

The performance of the deep learning model is highly dependent on the scale of the training data. However, model training and evaluation of the optimization results on large-scale data can be time-consuming. Consequently, evaluation methods are needed to make an approximate estimation. One popular method is estimating network performance with low fidelity results. For example, instead of using the full dataset with high resolution for architecture searching, a subset of the given dataset can be selected for architecture searching [17]. Also, training time and less filter can be applied in the searching stage [13].

The semantic segmentation of images is also an essential issue in computer vision. Its goal is to classify each pixel in the image. If image segmentation can be performed swiftly and precisely, it will be a significant step in areas like automatic driving, image beautification, and 3D reconstruction.

The first paper that successfully applied deep learning into image segmentation is fully convolutional networks (FCN) [18]. It introduced the fully convolutional network and applied deconvolution to map the feature map of the neural network to the original image size. In 2014, the first paper of Deeplab series Deeplab-V1 [9] came out. It has brought two essential methods: dilated convolution and fully connected conditional random fields (CRF) [19]. The dilated convolution can be considered as a traditional convolution with holes. It increases the receptive field without resolution reduction. The CRF is added in the final part of the network to improve details segmentation. U-Net [6] is also a classic segmentation model that performs feature fusion in a new way. Unlike FCN fusing features by element-wise addition, U-Net performs feature fusion by stitching in channel dimension. Another example is SegNet [11]. In 2017 and 2018, DeepV2 [9], DeepV3 [20] even DeepV3+ [8] have been proposed. One of the critical contributions is that it proposed an Atrous spatial pyramid pooling (ASPP) to obtain more robust segmentation results with multi-scale information. The PSPNet [21] also applied a similar idea to the network.

Another significant model is Mask R-CNN [22], which combines object detection and semantic segmentation. The Mask R-CNN has multiple branches for outputs of different tasks. The neural network learns two tasks at the same time and promotes each other. Furthermore, Mask R-CNN proposed RoiAlign to replace RoiPooling in Faster R-CNN [23]. The idea of RoiPooling is to map any piece of the input image to the corresponding area in the neural network feature map. RoiPooling utilizes a rounded approximation to discover the corresponding area, causing the correspondence to be offset from the actual situation. To solve this problem, instead of employing rounded approximation, RoiAlign applies linear interpolation to gain a more accurate corresponding area.

In the meantime, there are also some neural architecture search methods applied to image segmentation that have achieved superior segmentation results, especially for medical image segmentation. NAS-Unet [24] searches a U-like backbone network for medical image segmentation, and V-NAS [25] formulates the structure learning as differentiable neural architecture search, allowing the network to choose among 2D, 3D or Pseudo-3D (P3D) convolutions at each layer. Also, structures such as densely connected encoder-decoder CNN [26] are searched for medical image segmentation. Besides, there are also neural architecture search methods targeting 3D medical image, for example, SCNAS [27] with components of 3D convolution and 3D pooling.

III. METHODS

The key idea of the proposed method is by searching the downsampling layer, the bridge layer, and the upsampling layer with an recurrent neural network(RNN) controller, and the best neural network architecture targeting on image segmentation given learning data can be discovered by the AutoSegNet. Examples of the downsampling layer, the bridge layer, and the upsampling layer are shown in Fig. 2,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2995367, IEEE Access

Fig. 3 and Fig. 4, respectively. Meanwhile, whether to apply skip connections or not can also be searched by the RNN controller mentioned above. The skip connections include intra-cell connections and inter-cell connections.

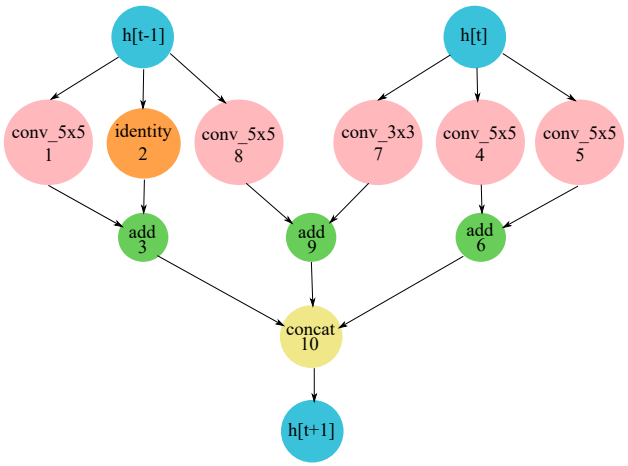


FIGURE 2: An example of searching the downsampling layer with three cells. $h[t-1]$ and $h[t]$ represent the two previous cells, and the number in the block represents the order of the operation.

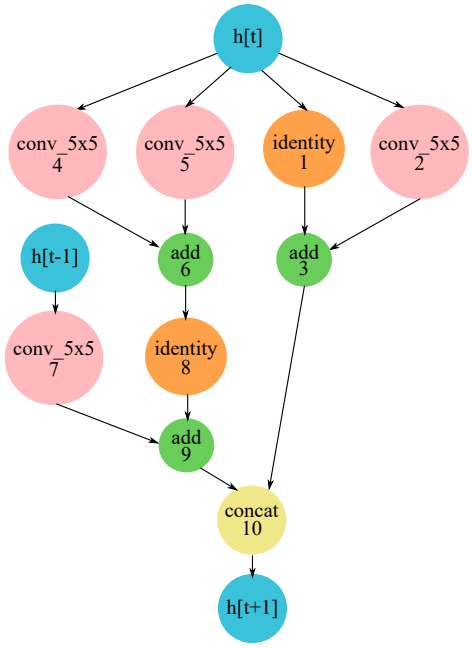


FIGURE 3: An example of searching the bridge layer with three cells. $h[t-1]$ and $h[t]$ represent the two previous cells and the number in the block represents the order of the operation.

Like ENAS [3], AutoSegNet includes two main components: an RNN controller for searching the network structure and a validation network for validating the searching

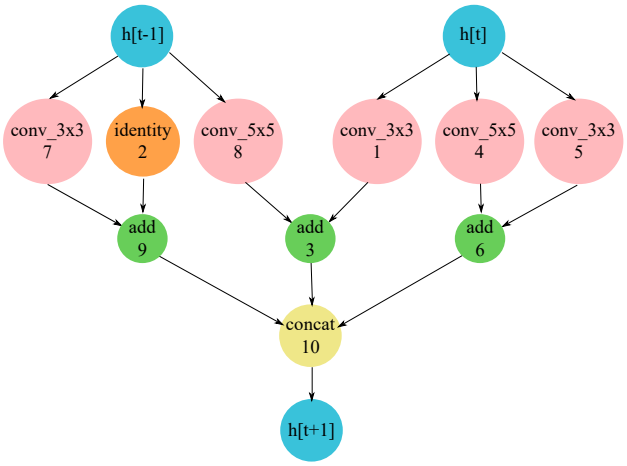


FIGURE 4: An example of searching the upsampling layer with three cells. $h[t-1]$ and $h[t]$ represent the two previous cells, and the number in the block represents the order of the operation.

architecture from the RNN controller. The RNN is a two-layer long short-term memory (LSTM) [28] network. The reason for choosing the RNN as a controller is that a variable-length output is needed from the controller. As the neural network architectures become different, the number of search parameters will be different. Since the output length of the RNN can vary, it is selected as the controller for neural network architectures searching. An example of how the RNN controller searches each layer is shown in Fig. 5. After the neural network architectures have been generated, the validation network will be applied to generate the network based on the search architectures, and then trains and validates the searched networks.

A. CONTROLLER

In AutoSegNet, the task of the RNN controller includes three parts: architecture search for the downsampling layer, architecture search for the bridge layer, architecture search for the upsampling layer, and architecture search for the long skip connection (short skip connection search is part of the layer search). As often noted, the choice of the layer structure of the previous layers may affect the choice of the succeeding layers. To better perform this, an LSTM module has been added to the RNN controller. The LSTM module records the state of the structure of the previous layers. Thus, when we select the layer components of the bridge layer, the structure of the downsampling layer is examined. Besides, when we determine the components of the upsampling layer, the structure of the downsampling layer and the bridge layer can also be acknowledged.

Due to the network structure (an encoder-decoder-based neural network), the searching order of the RNN controller is the downsampling layer, the bridge layer, the upsampling layer, and the long skip connection. The main reason is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2995367, IEEE Access

that the components of the later layers largely depend on the previous layers. Also, the long skip connection connects different layers of the neural network, which should be based on the whole network structure but not just parts of it. After selecting the components of the downsampling layer, the bridge layer, and the upsampling layer, whether the skip connection is needed will be decided by the RNN controller. Based on the information recorded by the LSTM module, the RNN controller will decide if skip connection is needed by a certain cell or layer. Since the hidden state of the LSTM module records the network structure information of the downsampling layer, the bridge layer, and the upsampling layer, the decision will be more reasonable. If the skip connection of a particular layer or certain cell is needed, the RNN controller will output value 1; otherwise, 0 will be the output. A structure of the RNN controller can be seen in Fig. 5.

For each layer, it might include several cells. A cell is a combination of several operations from the search space. The number of cells and operations of each cell are various, and they largely depend on the tasks or the given data. The specific task of RNN is to choose every operation of each cell based on the feedback of the validation network. The feedback is the accuracy of the result. For example, if each layer contains three cells, and each cell includes two operations, the RNN controller needs to choose a total of six operations for the layer. The encoder part is constructed by stacking the downsampling layer, and the decoder part is constructed by stacking the upsampling layer. An example of searched network architecture is shown in Fig. 7, which includes the downsampling layer, the hybrid dilated convolution layer, the bridge layer, the upsampling layers, and the skip connection. The detail architecture parameters of the downsampling layer, the bridge layer, and the upsampling layer are shown in Fig. 2, Fig. 3 and Fig. 4, respectively.

As the number of cells increases, the network will become more widespread, and the final neural network searched by the proposed network will appear similar to a GoogleNet [29]. The structures of layers based on the RNN controller shown in Fig. 5 can be found in Fig. 2, Fig. 3 and Fig. 4. By stacking these three types of layers, a completed neural network based on proposed methods is generated.

B. VALIDATION OF THE NETWORK

As mentioned above, based on the research, the mainstream network structures for image segmentation can be concluded as the encoder-decoder structure. An encoder learns the input images in different resolutions with a different receptive field while the decoder reconstructs the features learned from the encoder [6, 11]. For better representing the image features and decoding, additional optimization methods are introduced into the network, such as conditional random field (CRF) for better image details segmentation [9, 30], fusion of features in different scale using feature pyramids to merge shallow features and the global features [21] or dense skip connections to provide more shallow features [31].

Inspired by those state-of-the-art models, the main structure of the whole searching network is an encoder-decoder framework, a bridge layer connects the downsampling layers and the upsampling layers. The downsampling layers are considered as an encoder, which downsizes the spatial resolution of the input and develops a low-resolution feature map. The searched optimized network will be symmetric. This means that if the total number of the neural network layer is nine, there will be four downsampling layers, four upsampling layers, and a bridge layer connecting the downsampling layers and the upsampling layers.

During the downsampling phase, the neural network learns the input features from different receptive fields. However, to preserve the resolution, the hybrid dilated convolution is introduced. With a reduction of input size, some details of the original image might be lost. For example, when the input images reduce to 1/32 of the original image size, the original information with size 32×32 will disappear. Even with a decoder, it is hard to perform reconstruction from the given features. Consequently, as the size of the features reaches a specific value (for example, 16×16), a hybrid dilated convolution layer will replace the downsampling layer automatically for resolution preservation (as shown in Fig. 1). In this case, the searched neural network architecture turns into one of the state-of-the-art models [8].

The skip connections play an important role in image segmentation. Inter-cell skip connections (long skip connection) restore full spatial resolution while short skip connections can speed up the network convergence during the training stage [32]. As mentioned above, instead of manually adding skip connections (intra-cell and inter-cell) empirically, we prefer the RNN controller to select the skip connection by itself. The automation of the proposed method is maximized by reducing human disturbance. An example of the search neural network's architecture with long skip connections is shown in Fig. 6.

Three searched layers are shown in Fig. 2, Fig. 3, and Fig. 4, respectively. As can be seen from the figures, the searched layers are similar to the Inception block in GoogleNet [29]. The blocks in each figure are numbered (shown in the second line of the block). As discussed in the previous sections, each layer contains several cells. The proposed layers consist of three cells, and each cell includes two operations from the search space. For example, in Fig. 2, block one and block two belong to the first cell, and block four and block five belong to the second cell. Previous cells might be considered as the input of the next cell. For example, in Fig. 3, the first cell of the layer is part of the input of the second cell. Also, if a cell in a layer is not used as the input of other cells, it will be concatenated to become the final output directly. In Fig. 2 and Fig. 4, the input of all the cells of the layers is the output of the previous two layers. Thus, the output of the layer will be the concatenation of all the cells in the channel dimension.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2995367, IEEE Access

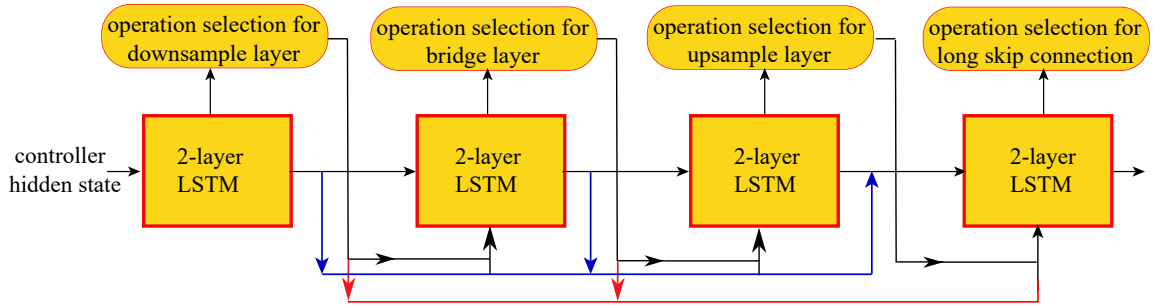


FIGURE 5: The structure of the RNN controller. The RNN controller is a 2-layer LSTM network. The operation selection will be based on the selection of the previous layer. The long skip connection search will be based on all previous layers.

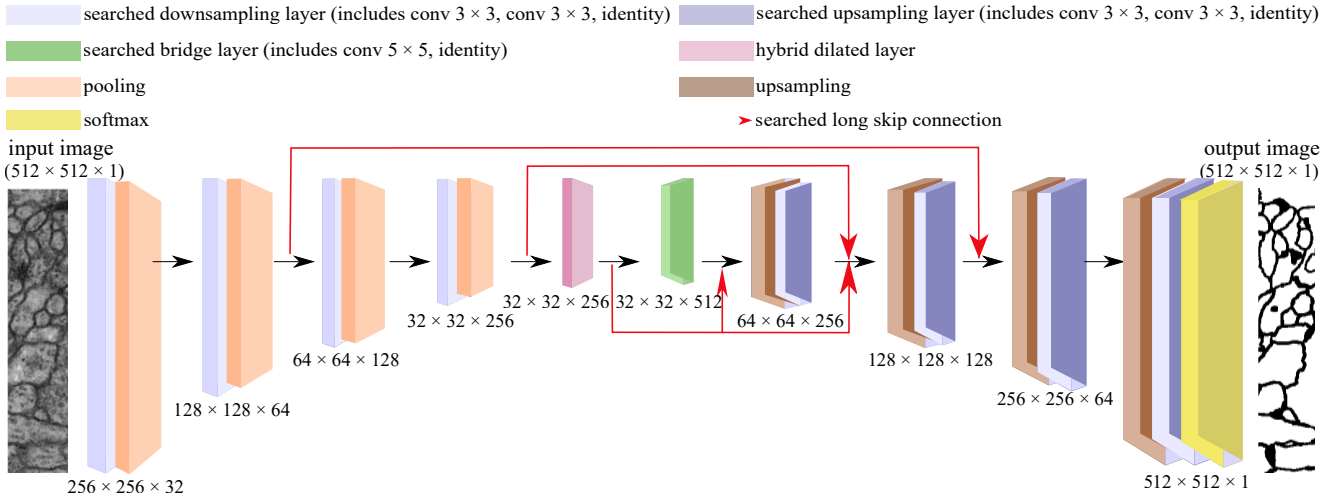


FIGURE 6: An example of a 10-layer neural network searched by the AutoSegNet with long skip connections. The shape of the input image is $512 \times 512 \times 1$ and the numbers shown under the blocks represent the shape of the output of that layer (*hight* $\times$ *width* $\times$ *channel*). The green blocks represent the downsampling layers, and the red block represents the hybrid dilated convolution layer, the purple block represents the bridge layer, and the blue blocks represent the upsampling layers while the red lines with an arrow represent the long skip connections searched by the RNN controller.

IV. EXPERIMENTS

The AutoSegNet was tested on the dataset of the industrial and medical areas. In this part, details and processing of the dataset are introduced. After that, the setting of the different datasets of AutoSegNet is explained. The experimental results are compared with other image segmentation models. Finally, the analysis will be given based on the results.

A. DATASET PREPARATION

1) Industrial dataset

The industrial dataset is a self-proposed dataset, which is from the components of the industrial products. The components might have some defects on the surfaces. Thus, AutoSegNet is used to highlight the defects. All the defects on the surface are highlighted manually in white while the others are in black.

The original number of images is 117, with an image size of 160×160. The images were cropped into a size of 32×32 and augmented to 5879 images (rotation, flipping, and pixels value scaling in a small range).

2) Medical dataset

The medical dataset is the 2D EM segmentation [33] data from the ISBI Challenge [34]. A full stack of EM slices has been used to search and train the neural network for segmentation. The training data only contains 30 sections with an image size of 512×512 from a serial section Transmission Electron Microscopy (ssTEM) dataset of the Drosophila first instar larva ventral nerve cord (VNC). These images represent the actual images in the real-world with small image alignment errors as well as noise. The labels given are fully annotated binary masks, in which pixels in white are the segmented objects, while pixels in black are the non-segmented objects.

As the number of the dataset is relatively small, image augmentation is needed. Augmentation includes flipping images from left to right, up to down, rotation with different angles as well as scaling of pixels value in a small range. After the augmentation, the total number of data increases from 30 to 1752.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2995367, IEEE Access

B. TRAINING DETAILS

In AutoSegNet, there are two sets of parameters that need to be trained, namely, the parameters of the RNN controller and the weight parameters of the validation network. Like ENAS, weight parameters are shared among all the child models. The stochastic gradient descent (SGD) is applied to minimize the loss function. For pixel-level segmentation, the standard cross-entropy loss is selected to compute on a minibatch of training data. While updating the controller parameters, the weight parameters of the validation network will be fixed.

Intending to maximize the automation of neural network searching, we reduce the number of manually input parameters to be as few as possible. The idea is quite straightforward. Each layer consists of several cells, and each cell includes operations searched from the search space. Typically, we fix the number of operations for a cell to be two and the number of cells per layer to be three. By doing so, we have a reasonable searching time and can also generate quality neural network structures.

Another critical parameter for the neural network architectures is the layer number of the searched network. As for the depth of the network, previous research says *"...We claim that most functions that can be represented compactly by deep architectures cannot be represented by a compact shallow architecture"* [31]. From this point of view, a more in-depth model means a better non-linear expression ability and can learn more complicated transformations, so that more complex features of the input can be fitted [35]. On the other hand, it does not mean a deeper network always performs better. A deeper network might bring problems like gradient vanishing or degeneration. Moreover, a more in-depth network could also overlearn the data and cause a vast computation cost.

The dataset given to the AutoSegNet might be various. They can be dataset from the medical area, industrial area, or natural images in a different size. If different settings are provided for different datasets, the degree of automation will be greatly reduced. Consequently, instead of giving the fixed layer number manually every time we train the AutoSegNet, we just let the network choose the layer number by themselves. We let the network downsample the input features to a fixed size. The input image size and minimum downsampling image size (or minimum downsampling ratio) decide the number of layers. It is not a part of the learning parameters of RNN but is directly computed when the input image size is given. For example, we set the minimum downsampling image size to be 16×16, and the minimum downsampling ratio to be $1:32$. The final minimum downsampling image size is the maximum value between the minimum downsampling image size and size downsampled by the maximum downsampling ratio. This is used to maximize the level of automation. For those who are not familiar with deep learning or neural network, it could be a difficult task to set the value of the layer number of the network. The proposed method tries to solve this problem. The encoder and decoder of the network is symmetric, which means the number of the downsampling layer and the upsampling layer will be the same in the searched network.

The smallest size by pooling may primarily be based on experience. Commonly, for image segmentation tasks, the smallest size by pooling is recommended to be $1/16$ or $1/8$ of the original image resolution [8, 20]. In both industrial and medical date sets, 75% of data is used as a training set, while 25% of data is test set. The final score is the average of three runs of training-testing. To test how the model complexity affects the performance of the network, we have excluded some components from the search space (3×3 hybrids dilated convolution with rates of 1, 2, and 3). The results show that both the Intersection over Union (IoU) scores on the self-proposed validation datasets and the visual effect of the AutoSegNet are worse than the current result.

Adam optimization algorithm [36], whose name is derived from adaptive moment estimation, is used to training the RNN controller with an initial learning rate of 0.0005. The shared weight parameters of the validation network are trained using SGD with an initial learning rate of 0.5. The total searching epochs were 200, and a factor of 0.98 decays the learning rate in every epoch after epoch 50. The norm of the gradient of weight parameters is clipped at 0.5. For preventing loss explosion, cosine annealing and SGD restart [37] are introduced to adjust the learning rate. Considering the generalization ability, we choose SGD as the optimizer of our network. Other optimizers, for example, the adaptive optimization algorithm, may exhibit a fast convergent rate at the initial stage of the training, but it may stagnate soon on the test set. Its generalization ability may be worse than that of the non-adaptive method [38, 39]. Although the adaptive optimization algorithm can show a fast convergence rate at the initial stage of training, its performance on the test set might soon stall. The searching stage stops automatically if they reach the *MAX_EPOCH* or meet the requirements of specific parameters such as *LOSS* and *ACCURACY*. The entire epoch for training the final searched architecture was 300.

C. RESULTS

The segmentation results of the industrial dataset and medical dataset are shown in this part. All the searched neural architectures are trained from scratch without pre-training. For the industrial dataset, the proposed method has cost of three and a half hours for searching the network structure on a single GPU (GeForce RTX 2080), while for searching the network architecture for the medical dataset, it takes eight hours and forty-five minutes on the same GPU. For the medical dataset with input image size 512×512, the number of segmentation network parameters is 41.93 million.

1) Experimental verification on the industrial dataset

The segmentation results for the industrial data are shown in Fig. 7. For each group, from left to right, they are the input image, the label, and the network prediction. As can be seen from the figures, most of the edges are well segmented.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2995367, IEEE Access

For example, in Figs. 7b-7d, the segmentation shape of the network outputs and labels are almost the same as well as the jagged edges. In Fig. 7a, the jagged edge cannot be segmented perfectly, which might be due to the discontinuity of the color edge. The discontinuity of the color might confuse the model to consider it as flaw. When the size of the training data is 2205, the average IoU score was 0.96 with the proposed method. While the training data size goes to 4414, the average Intersection over Union score was 0.9956 compared to a score of 0.9771 with the UNET trained with the same training data.

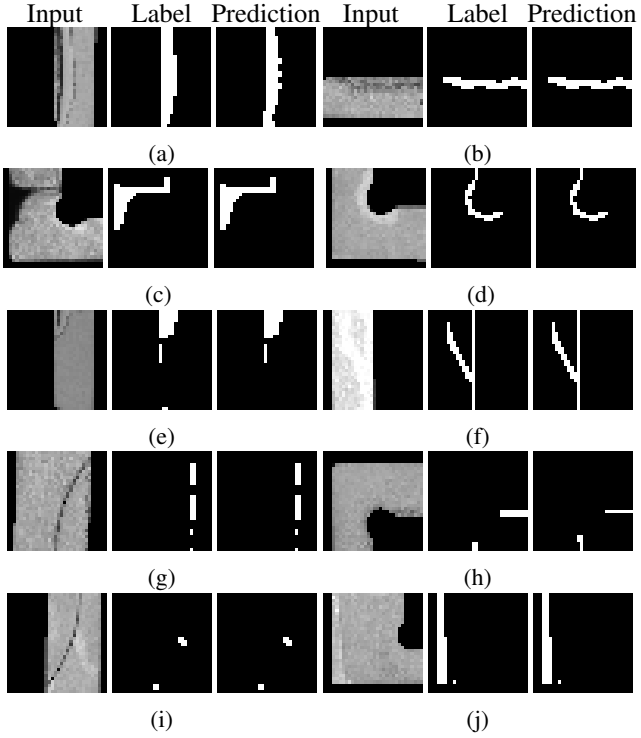


FIGURE 7: Segmentation results of the industrial dataset with flaw. For each sub-figure, from left to right, they are the input, the label, and the network prediction.

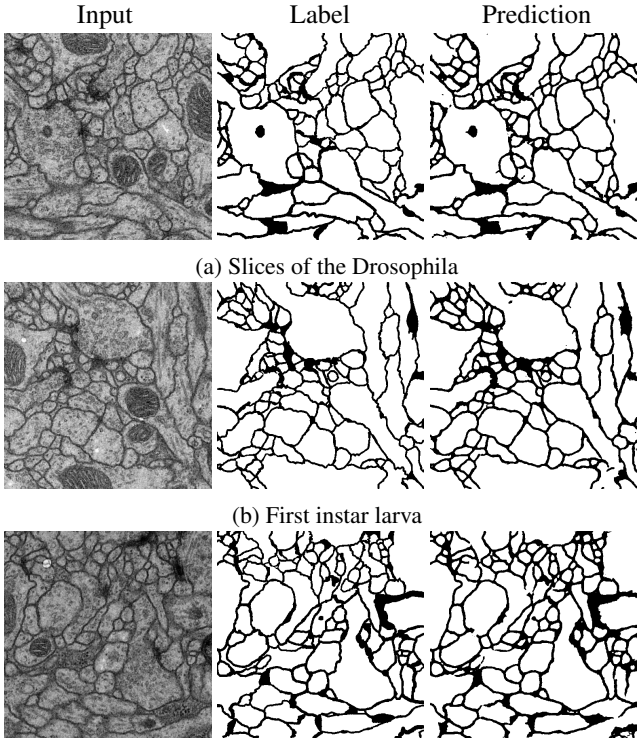
2) Experimental verification on the medical dataset

The network of the medical dataset was searched by an image size of 64×64 (by cropping). The searched layer structures (the downsampling layer, the upsampling layer, and the bridge layer) were transferred to train the original images with a size of 512×512, and the searching epochs of the medical dataset is 85. The segmented result of the medical dataset is shown in Fig. 8. Comparing the label and the network prediction, most of the edges of the prediction are clear and continuous. It indicates that the AutoSegNet can distinguish the edges and cell texture without treating them as edges. There are some black points inside the cell area in Fig. 8(a), especially around the edges. This is due to the dark areas near the edges, which might confuse the training model to consider it as edges. We also trained our model with different training size, from 657 to 1314. The IoU scores on the self-proposed validation set of the AutoSegNet are 0.9 and 0.9132, with training size 657 and 1314, respectively.

TABLE 1: IoU score of the validation on the medical dataset

Case number	Image size		IoU score
	Search	Train	
1	64×64	64×64	0.9051
2	64×64	512×512	0.9047

From Table 1, we can see that even though the neural network architecture was searched on images with a size of 64×64, it still generated qualified segmentation results on images with a size of 512×512. As the IoU score shows, the model trained with image size of 512×512 is 0.0004 lower than the the model trained with image size of 64×64. This reveals the robustness of the searching architectures.



(a) Slices of the Drosophila

(b) First instar larva

(c) Ventral nerve cord

FIGURE 8: Segmentation results of the medical dataset. For each sub-figure, from left to right, they are the input, the label, and the network prediction. Sub-figures (a) to (c) represent the slices of the Drosophila, first instar larva and ventral nerve cord, respectively. In (a), there are some black points inside the cell area, especially around the edges. It is because there are some dark areas near the edges, which might confuse the training model to consider it as edges.

The segmentation results on the industrial and medical datasets of Fig. 7 and Fig. 8 indicate that the AutoSegNet can generate neural network architectures with a variety of given learning datasets. From the network prediction, we can see that continuous edges are well-segmented. Besides, even though the neural network architectures are searched

in a low fidelity dataset, it still generated qualified results when we apply them to high fidelity datasets, which shows the robustness and the capability of the generalization of AutoSegNet.

TABLE 2: Comparison of the IoU score on the self-proposed validation set. The PSPNet is used for comparison [40].

Network	Mean IoU Score	
	Industrial dataset	Medical dataset
UNET	0.9771	0.9119
PSPNet	0.9504	0.8975
AutoSegNet	0.9956	0.9132

We also compared the mean IoU of the proposed AutoSegNet and UNET [6] and PSPNet [21] which is the state of the art. As shown in Table 2, the mean IoU scores on the self-proposed validation set of the AutoSegNet are 0.9956 and 0.9132 with the industrial and medical datasets, respectively. While with the UNET, the scores are 0.9771 and 0.9119, which indicates that the AutoSegNet shows a more superior efficiency. And when compared the mean IoU score with PSPNet, for the Industrial dataset, the score of the proposed method is also higher than that of PSPNet in both industrial dataset and medical dataset.

V. CONCLUSION

In this paper, we proposed an efficient AutoSegNet for image segmentation, especially for the industrial and medical datasets. We maximized the level of automation of the AutoSegNet. Compared to the previous searching neural architectures in which parameters such as layer number, cell number, and block number are required, the AutoSegNet requires nothing other than the learning data. Besides, compared with other NAS methods, the AutoSegNet holds a small search space. Meanwhile, it covers the functions of most typical image segmentation neural networks. These characteristics account for its significant efficiency. Furthermore, searching the neural network with a lower fidelity dataset, and applying it to a higher fidelity dataset, the AutoSegNet still generates outstanding segmentation results, which indicates its robustness and generalization.

REFERENCES

[1] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," arXiv: 1611.01578, 2016.

[2] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," ArXiv: 1711.00436, 2018.

[3] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in Proceedings of the 35th International Conference on Machine Learning, vol. 80, 2018, pp. 4095–4104.

[4] H. Cai, J. Yang, W. Zhang, S. Han, and Y. Yu, "Path-level network transformation for efficient architecture search," in International Conference on Machine Learning, 2018, pp. 678–687.

[5] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," ArXiv: 1611.02167, 2016.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Lecture Notes in Computer Science, 2015, pp. 234–241.

[7] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1925–1934.

[8] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in European Conference on Computer Vision, 2018, pp. 833–851.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834–848, 2014.

[10] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in Winter conference on applications of computer vision (WACV), 2018, pp. 1451–1460.

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

[12] A. Krizhevsky, G. Hinton et al., "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep. 4, 2009.

[13] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in Conference on Computer Vision and Pattern Recognition, 2018.

[14] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 82–92.

[15] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The penn treebank," in Proceedings of the workshop on Human Language Technology, 1994, pp. 114–119.

[16] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.

[17] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast bayesian optimization of machine learning hy-

perparameters on large datasets," in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017), vol. 54, 2017, pp. 528–536.

[18] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640–651, 2017.

[19] J. D. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 282–289.

[20] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," arXiv:1706.05587, 2017.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6230–6239.

[22] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.

[23] R. Girshick, "Fast R-CNN," in International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448.

[24] Y. Weng, T. Zhou, Y. Li, and X. Qiu, "NAS-Unet: Neural architecture search for medical image segmentation," IEEE Access, vol. 7, pp. 44 247–44 257, 2019.

[25] Z. Zhu, C. Liu, D. Yang, A. Yuille, and D. Xu, "V-NAS: Neural architecture search for volumetric medical image segmentation," ArXiv: 1906.02817, 2019.

[26] M. Aliasghar and B. Ulas, "Automatically designing CNN architectures for medical image segmentation." Machine Learning in Medical Imaging, vol. 11046, pp. 98–106, 2018.

[27] S. Kim, I. Kim, S. Lim, W. Baek, C. Kim, H. Cho, B. Yoon, and T. Kim, "Scalable neural architecture search for 3D medical image segmentation," ArXiv: 1906.05956, 2019.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.

[30] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1529–1537.

[31] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1175–1183.

[32] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in Deep Learning and Data Labeling for Medical Applications, 2016, pp. 179–187.

[33] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomancak, and V. Hartenstein, "An integrated micro- and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy," PLoS Biology, vol. 8, no. 10, p. e1000502, 2010.

[34] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann, T. Liu, M. Seyedhosseini, T. Tasdizen, L. Kamentsky, R. Burget, V. Uher, X. Tan, C. Sun, T. D. Pham, E. Bas, M. G. Uzunbas, A. Cardona, J. Schindelin, and H. S. Seung, "Crowdsourcing the creation of image segmentation algorithms for connectomics," Frontiers in Neuroanatomy, vol. 9, p. 142, 2015.

[35] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, "On the expressive power of deep neural networks," in Proceedings of the 34th International Conference on Machine Learning, vol. 70, 2017, pp. 2847–2854.

[36] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," International Conference on Learning Representations, 12 2014.

[37] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with restarts," ArXiv: 1608.03983, 2016.

[38] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in Advances in Neural Information Processing Systems, 2017, pp. 4148–4158.

[39] N. S. Keskar and R. Socher, "Improving generalization performance by switching from adam to SGD," ArXiv: 1712.07628, 2017.

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

ZHIMIN XU received the B.S. and M.S. degrees in electronic engineering from Fudan University, China, in 2005 and 2008, and the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Hong Kong, in 2012. He is currently the founder and CEO of SharpSight Limited. His research interest includes computational imaging, light field, deep learning, and computer vision.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2995367, IEEE Access

SI ZUO received the B.S. degree in electrical engineering from Jiangxi University of Finance and Economics, China, in 2016 and the M.S degree in electrical engineering from Aalto University, Finland, in 2018. She is currently an algorithm engineer at SharpSight Limited. Her main research interest includes deep learning and computer vision.

EDMUND Y. LAM (M'00–SM'05–F'15) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University. From 2010 to 2011, he was a Visiting Associate Professor with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. He is currently a Professor in electrical and electronic engineering with The University of Hong Kong, where he is also the Director of the Computer Engineering Program. His main research interest includes computational imaging. He is a Fellow of OSA, SPIE, IS&T, and HKIE. He was a recipient of the IBM Faculty Award.

BYOUNGHO LEE (M'94–SM'00–F'14) is currently a Professor in the Department of Electrical and Computer Engineering, Seoul National University, South Korea. He is a fellow of IEEE, SPIE and the Optical Society of America. He is a member of the Korean Academy of Science and Technology and a senior member of the National Academy of Engineering of Korea. He served on the Board of Directors of OSA. He has received many awards, including the Jin-Bo-Jang Medal from the President of South Korea in 2016. He has served as the vice President of the Korean Information Display Society. He is currently the President of the Optical Society of Korea. His research interest includes 3D and augmented reality display and imaging systems.

NI CHEN received the B.S. degree in software engineering from Harbin Institute Technology University, China, in 2008, the M.S. and Ph.D. degrees in electrical engineering from Chungbuk National University and Seoul National University, Korea, in 2010 and 2014, respectively. She is currently a research assistant professor in the Department of Electrical and Computer Engineering at Seoul National University, Seoul, Korea. From 2014 to 2016, she was a Research Scientist with the University of Hong Kong, and from 2016 to 2017, and she was an Associate Professor with the Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences. Her research interest includes three-dimensional optical imaging and display.