Performance Evaluation of Semantic Segmentation Using Different Encoder–Decoder Architectures

DENG-YUAN HUANG[1]*, JING-YAN YANG[1] and YU-YUN WANG[2]

[1]*Department of Computer Science & Information Engineering, Da-Yeh University,*
[2] *Department of Electrical Engineering, Da-Yeh University*
No. 168, University Rd., Dacun, Changhua 51591, Taiwan, R. O. C
**kevin@mail.dyu.edu.tw*

ABSTRACT

Semantic segmentation, also known as dense prediction, is a task in computer vision that is critical for scene understanding and commonly used for pixel-wise labeling of whole images. This paper proposes a new semantic segmentation network architecture dedicated to decoder structural design. Moreover, different mapping mechanisms are introduced as a part of the decoder network. The proposed architecture was tested on the 21 classes of the Pascal Visual Object Classes Challenge 2012 data set. The proposed method has a higher mean intersection over union score and pixel accuracy than those of the U-SegNet, UNet, and ENet models but similar results to those of the SegNet model. Additionally, the effect of using magnification methods in the decoder network on object segmentation performance was investigated.

***Key Words*: Semantic segmentation, Scene understanding, Encoder network, Decoder network**

不同編解碼器架構下語義分割之性能評估

黃登淵 [1]*　楊靖言 [1]　王喻筠 [2]

[1] 大葉大學資訊工程學系
[2] 大葉大學電機工程學系
51591 彰化縣大村鄉學府路 168 號
**kevin@mail.dyu.edu.tw*

摘要

在計算機視覺中，語義分割是最重要的任務之一，因其對於場景理解至關重要。通常，語義分割用於整個影像之像素標記，故又稱為密集預測。本文之目的在於提出一種新的語義分割網路架構，特別著重在解碼器結構的細部設計。本文創新之處在於引入不同的映射機制作為解碼器網路的一部分。本文所提方法在 21 個類別的 Pascal VOC 2012 資料集上進行測試，其結果同時與 U-SegNet，UNet 和 ENet 等模型進行比較，實驗結果顯示，本文所提方法具有更高的 mIoU 分數與像素精度，但與 SegNet 相比，其分割效能相當。此外，本文亦探究解碼器網路中特徵圖放大方法對物體分割性能之影響。

關鍵詞：語義分割，場景理解，編碼器網路，解碼器網路

I. INTRODUCTION

The task of image semantic segmentation is to classify instances in a whole image at the pixel-level, each instance (or class/category) corresponding to an object in the image or representing a part of the image, namely person, bicycle, grass, road, sky, and so on. This task is also referred to as dense prediction. Hence, the goal of this task is to label each pixel in an image with a corresponding class of what is being represented. Semantic segmentation (or object segmentation) is very critical for scene understanding to realize how a deep learning model can better learn the global context of a visual content.

In past decades, artificial intelligence (AI) based on deep learning has received a lot of attention, and it successfully attracts many important investments in the fields of self-driving vehicles [21], robotic arms [17], and medical diagnosis [9]. More and more applications require accurate and efficient image segmentation techniques. There are several key problems in the field of computer vision [2], namely image classification [3], object detection [18] and semantic segmentation [4]. Among them, image classification [3] is to identify and label the class of each object in an entire image. Object detection [18] is to identify and locate all objects existing in the image. Semantic segmentation [4] is to identify the category of each pixel in the image, which usually labels same category pixels with the same color, as shown in Fig. 1.

Semantic segmentation is one of the high-level tasks in computer vision that provides a very important way towards complete scene understanding through machine perception to visual context [15]. In the application of autonomous driving, vehicle behavior such as the determinations of steering angle and forward speed can be predicted by semantic segmentation of road scenery [24]. As a core task of computer vision, the importance of scene understanding can be inferred by the increasing number of applications on image segmentation. With the popularity of deep learning in past years, many semantic segmentation problems are being tackled using deep neural network architectures, most often convolutional neural networks (CNNs), which outperform other methods by a wide margin in terms of accuracy and efficiency [1, 4, 14, 20].

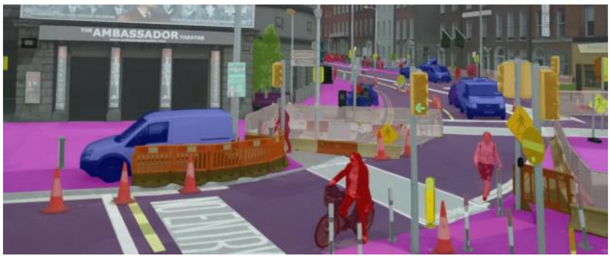

Fig. 1. Typical results of semantic segmentation. In this picture, red, blue, green, yellow, purple and gray colors are used to label pedestrian, car, plants, traffic sign, sidewalk, and building, respectively (picture courtesy of [11]).

Fully convolutional network (FCN) [14] has been the first to develop an end-to-end architecture for semantic segmentation. The FCN uses images of any size as input and creates segmented images with the same size. The authors first modified the popular CNN's architectures such as AlexNet [10], VGG16 [22], and GoogLeNet [23] to have a variant size input while replacing all fully connected layers with convolutional layers. Because the network generates multiple feature maps with small sizes and dense representations, upsampling is required to produce an output of the same size with the input. Generally, upsampling consists of a convolutional layer with a stride greater than one. It is often called deconvolution because it builds an output size that is larger than the input. In this way, the network is trained using pixel-by-pixel loss. In addition, they added skip connections in the network to incorporate high-level feature representations with more specific and dense representations at the top of the network. This work has reached a 62.2% mIoU score on the 2011 PASCAL VOC segmentation challenge using pre-trained models on the ImageNet dataset.

UNet [20] is an extension of the FCN proposed by J. Long et al. in 2015 [14], which comprise two parts: an encoding part to compute features and a decoding part to spatially localize patterns in the image. The downsampling or encoding part has a FCN-like architecture extracting features with 3×3 convolutions. The upsampling or decoding part uses transposed convolution (or deconvolution) reducing the number of feature maps while increasing their sizes. Feature maps from the downsampling part of the network are copied to the corresponding upsampling part to avoid losing pattern information. Finally, a 1×1 convolution layer is used to deal with the feature maps to produce a segmentation map and thus classifies each pixel of the input image into categories. As consequence, the number of parameters of the model is reduced and it can be trained with a small labelled dataset.

Mask R-CNN architecture was proposed by K. He et al. [4] in 2017 and it beat all earlier benchmarks on many COCO challenges. The Mask R-CNN is a Faster R-CNN [19] with three output branches: the first layer is a region proposal network (RPN) used to extract the region of interest (RoI). The second one processes the RoI to generate feature maps that are directly used to compute the bounding box coordinates and the predicted class. The third layer (i.e., FCN) is used to tackle feature maps to create the binary mask for a given RoI with a fixed size. The success of Mask R-CNN lies on the introduction of multi-task loss that involves the losses of the bounding box coordinates, the predicted class and the segmentation mask. This model has obtained 37.1% and 41.8% average precision score on the 2016 and 2017 COCO segmentation challenges, respectively.

SegNet was originally submitted to CVPR2015 but it is not being published in CVPR. Instead, it is published in 2017 TPAMI [1]. SegNet is a combination of encoder and decoder architecture with a final pixelwise classification layer. At the encoder network, same as the literature [20], convolutions and max pooling are carried out. While doing 2×2 max pooling, the corresponding max pooling indices are stored. At the decoder network, unpooling is conducted, where the max pooling indices at the corresponding encoder layer are recalled to upsample the feature maps. Finally, a K-class softmax classifier is used for pixelwise class prediction. Experimental results on CamVid dataset for road scene segmentation show 71.2% class average accuracy.

U-SegNet, based on the architecture of SegNet [1], was proposed by Kumar et al. [11] in 2018, where the feature maps are magnified by the so-called unpooling method that passes the pooling indices to the corresponding upsampling layers in the decoder. U-SegNet adopts the similar structure of the UNet [20] by adding the skip-connection structure (i.e., SC structure) in the high-level convolutional layers, which can help to capture abstract information, thereby improving the resulting semantic segmentation. U-SegNet is mainly used for brain image segmentation tasks, and its segmentation performance is better than the state-of-the-art methods such as SegNet and UNet.

Some popular deep convolutional networks such as AlexNet [10], VGG16 [22], GoogLeNet [23], and ResNet [5], which won the first place in 2012, 2013, 2014, and 2015 ImageNet competitions, respectively, have made significant contributions to the field of computer vision, as they are often adopted as the basis of semantic segmentation (or object segmentation) networks. Especially, the introduction of residual blocks in ResNet [5] has contributed to deeper neural networks with 152 layers. The residual blocks resolves the difficulty of training a really deep architecture by introducing identity skip connections so that information in input layer can be passed to the next layer. The introduction of residual blocks becomes an important driving force for improving the performance of state-of-the-art semantic segmentation networks.

In general, semantic segmentation not only requires categorization at pixel level but also a mechanism to project the discriminative features learned at different stages in the encoder onto the feature maps of the corresponding stages in the decoder. Different semantic segmentation approaches use different mapping mechanisms as a part of the decoder network. It turns out that the design of decoder network in the domain knowledge of semantic segmentation is a core problem involving the performance of that network. Therefore, the goal of this paper is to propose a novel architecture of semantic segmentation network dedicating to the design of the decoder using different feature mapping mechanisms. Comparison results with existing approaches show the feasibility and effectiveness of the proposed decoder network.

The rest of this paper is organized as follows: Section 2 describes the framework of the proposed method of object segmentation (or semantic segmentation). Experimental results are given and discussed in Section 3. Finally, the concluding remarks are provided in Section 4.

II. THE PROPOSED SEMANTIC SEGMENTATION ARCHITECTURE

The proposed architecture of semantic segmentation network is shown in Fig. 2. There are a total of 5 stages in the encoder network. For the same stage, the feature maps of each convolutional layer output have the same size and channels. In the stages of 1-5, they contain 2-2-3-3-3 layers of Conv2D block, and the number of channels is 64-128-256-512-512 channels, respectively. For the convenience of describing the encoder network, each convolutional layer is numbered and denoted as EC-Conv-m-n, where EC, m and n represent Encoder, Stage and Layer, respectively, so EC-Conv-1-1 is represented as the first layer in the first stage of the encoder network.

In this work, all convolutional layers use a 3×3 convolution kernel with learnable weighted parameters. The pooling layer uses MaxPooling to reduce the feature map. After Conv2D layer, batch normalization [7] is applied to normalize the data of the convolutional layer output to avoid the occurrence of gradient vanishing or explosion in the backpropagation, followed by the use of PReLU (Parametric Rectified Linear Unit) [6], which is a nonlinear activation function that keeps the positive values unchanged but outputs smaller negative values instead of setting them to zero for the negative values. In this work, one important consideration using the PReLU activation function is to avoid gradient vanishing, and another is to avoid destroying the features output by the convolutional layers.

Now, we turned our attention to the details of the decoder network, where each stage corresponds to the same one in the encoder network. In the decoder network, each convolution layer is denoted as DC-Conv-m-n, where DC, m, and n represent decoder, stage, and layer, respectively. For the design of semantic segmentation architectures, most of the encoder networks are the same. The only difference exists in the decoder network. In this work, we modified the decoder network of SegNet [1] by adding the skip connection (SC) structure. This idea is inspired by the feature pyramid network proposed by Lin et al. [13], from which it confirms the better result of pixel prediction accuracy when using the SC structure.

To clearly describe the detail structure of SC in our work, we took stage-4 as an example for illustration. In this case, we chose the last layer of stage-4 in the encoder network (EC-Conv-4-3) because the deepest layer in the same stage can extract the most discriminative features. Then, we select the Upsampling-4 layer of the corresponding stage in the decoder network. In our case, image magnification by a factor of two is implemented using the upsampling method with nearest neighbor. These two layers, i.e., EC-Conv-4-3 and Upsampling-4, are then added together pixel by pixel. Finally, a 3×3 convolutional kernel is applied to generate the layer DC-Conv-4-1, as shown in Fig. 3(a).

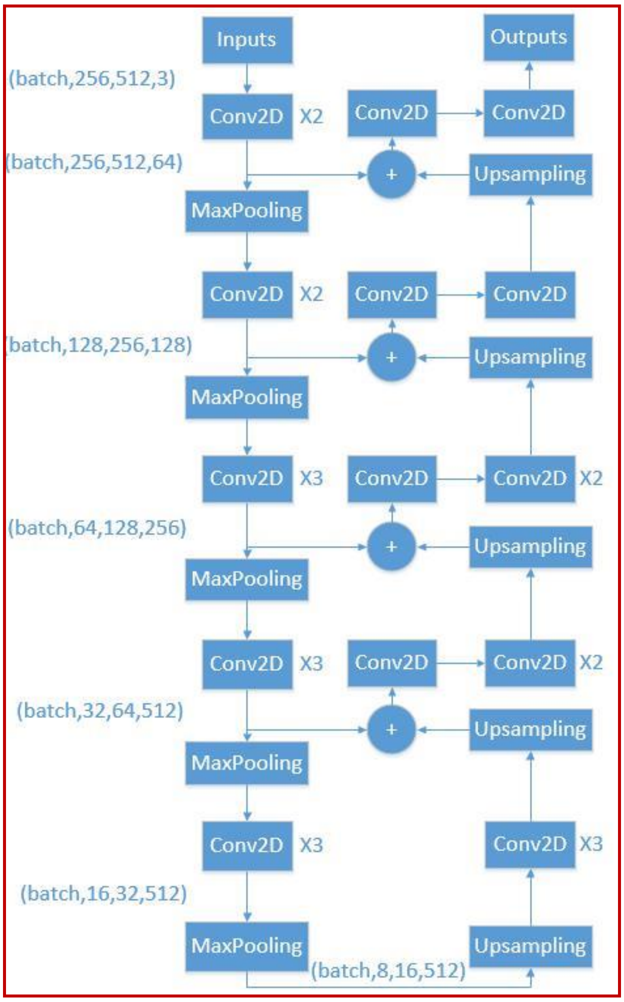

Fig. 2. The proposed semantic segmentation architecture with a symmetric encoder and decoder structure. As shown in this figure, the symbols ×2 and ×3 indicate repeated two and three times of Conv2D block, respectively. In the parentheses of (batch, height, width, channel), they represent batch size, image height, image width, and channel number in order, respectively.

This work compares the differences of the decoder network between ours, SegNet [1], UNet [20] and ENet [16]. As shown in Fig. 3(b), the MaxUnpooling can be considered as a reverse of MaxPooling but it recovers the feature maps resolution to its input size by recalling the max-indices from the corresponding layer in the encoder network. As shown in Fig. 3(c), UNet uses transposed convolution (or deconvolution) to first enlarge the feature maps by a factor of two and then convolve with a 3×3 kernel. Subsequently, these convolution layers are stacked (or concatenated) instead

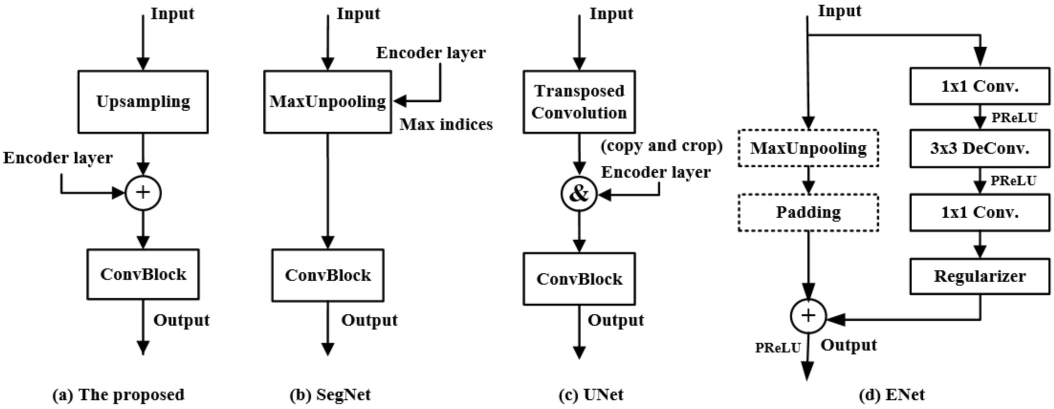

Fig 3. Difference of magnification methods in the decoder. (a) The proposed, (b) SegNet [1], (c) UNet [20], and (d) ENet [16].

of addition like ours with the layers corresponding to the same stage in the encoder network. But for ENet, the magnification of feature maps in the decoder network is realized using the MaxUnpooling method, where the maximum indices are stored in the encoder network, as shown in Fig. 3(d). For instance, the maximum indices in bottleneck2.0 are recorded and later used for the enlargement of feature maps in bottleneck4.0. The use of bottleneck in ENet is heavily inspired by the idea of the ResNet [5].

In our work, the last layer in the decoder network is used to predict pixel class using softmax function. To train the proposed semantic segmentation network, the loss function of pixel-wise cross entropy is used and defined as in Eq. (1).

$$Loss = -\frac{1}{C}\sum_{y=0}^{H-1}\sum_{x=0}^{W-1} p(x,y)\log t(x,y), \tag{1}$$

where C, H, and W denote the number of pixel classes, height and width of feature maps, respectively. $p(x,y)$ and $t(x,y)$ are the predicted and true class of pixel (x,y) in test image, respectively.

III. RESULTS AND DISCUSSION

The proposed method of semantic segmentation is evaluated on the Pascal VOC 2012 dataset. The operation system is Linux Ubuntu 18.04.2 and the integrated development environment is Python 3.5.2 64-bit version with installed library packages of Keras 2.2.4, Tensorflow GPU version 1.13.1 and Opencv-python 3.2.0.8. The system runs on NVIDIA DGX station with an Intel Xeon E5-2698 v4 2.2GHz processor and 256GB memory. The DGX station is equipped with 4x Tesla V100 GPU cards and each with 32GB memory. Thus, the total amount of GPU memory is 128 GB.

The Pascal VOC 2012 dataset is popular and often used for object detection and segmentation. Over 11k images form the training and validation dataset, while 10k images are dedicated to the test dataset. The metric of mean Intersection over Union (mIoU) is commonly used to evaluate segmentation results. As for the metric of IoU, it is also used in object detection to evaluate the relevance of the pixel prediction. The IoU is the ratio between the areas of overlap and union between the ground truth and the predicted areas. Moreover, the mIoU is the average between the IoU of the segmented objects over all the images in the test dataset.

In this work, we used Keras and TensorFlow to implement the proposed semantic segmentation network architecture. The proposed architecture is trained and tested on the Pascal VOC 2012 dataset. The results of our proposed semantic segmentation architecture are compared with those obtained by UNet [20] and ENet [16]. The commonly used metrics for semantic segmentation evaluation are pixel accuracy (PA) and mIoU as defined in Eq. (2) and Eq. (3), respectively.

$$\mathrm{PA} = \frac{\sum_{i=0}^{C} p_{ii}}{\sum_{i=0}^{C}\sum_{j=0}^{C} p_{ij}}, \qquad (2)$$

and

$$mIoU = \frac{1}{C+1}\sum_{i=0}^{C}\frac{p_{ii}}{\sum_{j=0}^{C} p_{ij} + \sum_{j=0}^{C} p_{ji} - p_{ii}}, \qquad (3)$$

where C is the number of total classes to be predicted. Note that the background is also considered in Eq. (2) and Eq. (3), indicating that the total classes are C+1, where we can see the indices of i=0 to C and j=0 to C. The symbol p_{ii} means that the pixel (x, y) belongs to class i and it is also identified as class i. However, p_{ij} can be considered as false negative because the pixel belonging to class i is identified as class j, and p_{ji} can be thought as false positive due to the j-class pixel erroneously identified as class i. Finally, the Adam optimizer [8] is applied to train the proposed architecture. The learning rate and learning attenuation value are set to 5×10^{-4} and 2×10^{-4}, respectively.

Table 1 shows the results of mIoU and pixel accuracy for different semantic segmentation architectures. As seen from this table, the proposed method has higher mIoU and pixel accuracy when compared with the results achieved by UNet, ENet and U-SegNet, but has comparable results with SegNet. The improved rates of the proposed method in both mIoU and pixel accuracy are 3.11% ((45.29-43.92)/43.92*100%) and 0.6% ((84.81-84.30)/84.30*100%), respectively when compared with the UNet. The visual results of object segmentation for ours, UNet, ENet and U-SegNet are shown in Fig. 4.

As observed from Table 1, the segmentation performance in terms of mIoU and pixel accuracy for the U-SegNet is quite poor when compared with all the methods. U-SegNet is a deep learning architecture by adding the SC structure only in high-level layers, which is dedicated to the segmentation task of brain images. Aiming at the small medical image database IBSR-18, U-SegNet shows good performance in the classification rate of magnetic resonance images (MRI) of brains into 4 categories. However, the performance on the Pascal VOC 2012 dataset are not satisfactory. It might be due to the use of less convolutional layers and only uses the SC structure in the high-level layers when compared with the models of the proposed, SegNet, UNet, and ENet. U-SegNet only uses the high-level layers' SC structure because they believed it (i.e., SC structure) can help in reducing random noise in the low frequency for the images of IBSR-18 dataset. Clearly, this consideration cannot be applied to the Pascal VOC 2012 dataset.

Table 1. Results of mIoU and pixel accuracy on the Pascal VOC 2012 dataset for different semantic segmentation architectures

Variants	mIoU	pixel accuracy
Proposed	**45.29**	84.81
SegNet	45.08	**84.85**
UNet	43.92	84.30
ENet	29.60	78.88
U-SegNet	0.08	0.74

Note: The values in bold means the highest in the mIoU and pixel accuracy columns.

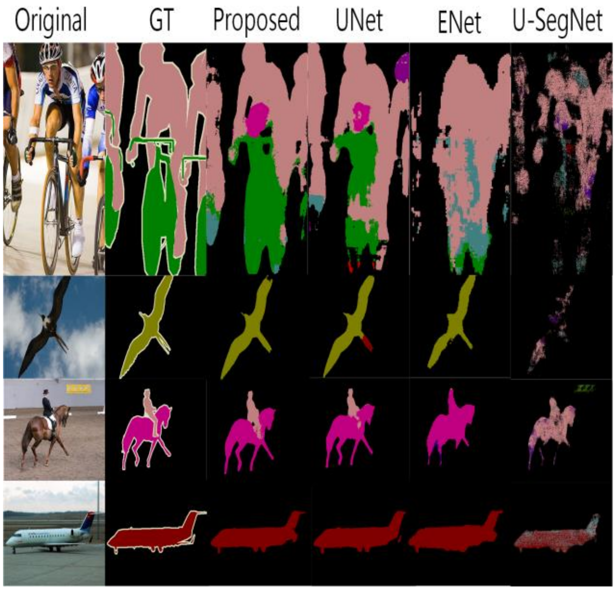

Fig. 4. Visual results of object segmentation for ours, UNet, ENet, and U-SegNet. As observed from this picture, the segmentation results obtained by ours clearly outperform those obtained by UNet, ENet, and U-SegNet.

Table 2 further shows the effect of different magnification methods in the decoder network on the performance of object segmentation for the proposed architecture. As observed from this table, the Upsampling method with nearest interpolation has better results either in terms of mIoU or pixel accuracy when compared with the Unpooling method. Clearly, the improved rate of mIoU is more prominent than pixel accuracy.

Table 2. Results of mIoU and pixel accuracy on the Pascal VOC 2012 dataset for different magnification methods in the decoder network for the proposed architecture

Methods	mIoU	pixel accuracy
Unpooling	43.72	84.43
Upsampling	**45.29**	**84.81**

Note: The values in bold means the highest in the mIoU and pixel accuracy columns.

Table 3. Results of 21 categories of IoU scores in the Pascal VOC 2012 dataset for the proposed, UNet and ENet

category	Proposed	UNet	ENet
background	**85.72**	85.49	81.30
aeroplane	66.52	**67.39**	29.82
bicycle	27.05	**27.20**	10.13
bird	**48.25**	45.55	27.83
boat	**39.68**	35.32	7.05
bottle	27.55	**28.21**	4.98
bus	61.55	**62.76**	47.28
car	**62.95**	57.11	51.64
cat	49.53	**51.08**	38.49
chair	**16.37**	13.86	n/a
cow	**41.38**	41.05	30.80
diningtable	**30.20**	22.44	20.85
dog	**44.49**	39.66	28.76
horse	**41.57**	38.28	31.96
motorbike	55.47	**57.83**	38.99
person	**62.11**	61.63	52.18
potted-plant	21.56	**24.78**	8.95
sheep	**48.28**	44.22	31.92
sofa	21.35	**21.57**	14.45
train	**60.03**	58.59	39.24
tvmonitor	**39.52**	38.23	24.94

Note: The values in bold means the highest scores of IoU for a specific class.

Table 3 shows the results of 21 categories of IoU scores in the Pascal VOC 2012 dataset for the proposed, UNet and ENet. As seen from this table, the proposed architecture has 13 categories with the highest IoU scores when compared with the results obtained by UNet and ENet, indicating the superiority of the proposed method. As revealed by this table, ENet has the worst results among the three networks. As inferred from its network architecture, it might be caused by the less convolution layers and asymmetrical encoder-decoder architecture.

IV. CONCLUSION

In this paper, we have proposed a symmetric encoder and decoder semantic segmentation network architecture. The novelty of this architecture is to introduce different mapping mechanisms as a part of the decoder network. The proposed architecture was tested on the Pascal VOC 2012 dataset and satisfactory results are achieved. The proposed method has higher mIoU score and pixel accuracy when compared with the networks of U-SegNet, UNet and ENet but comparable with the results of SegNet. The improved rates of the proposed method in mIoU score and pixel accuracy are 3.11% and 0.6%, respectively, when compared with the UNet. Moreover, the effect of magnification methods in the decoder network on the performance of object segmentation is also investigated. As compared with the Unpooling method, the Upsampling method with nearest interpolation has better results either in terms of mIoU score or pixel accuracy. In addition, the improved rate of mIoU score is more prominent than the metric of pixel accuracy. In the future work, more variants of decoder network will be investigated to further improve the performance of object segmentation.

V. ACKNOWLEGEMENT

This work is completely supported by a grant from Ministry of Science and Technology (MOST), Taiwan, under contract MOST-107-2221-E-212-012 and MOST-108-2221-E-212-012.

REFERENCES

1. 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(12), 2481-2495.
2. Bradski, G., A. Kaehler and V. Pisarevsky (2005) Learning-based computer vision with intel's open source computer vision library, *Intel Technology Journal*, 9(2), 119-130.
3. Cireşan, D., U. Meier and J. Schmidhuber (2012) Multi-column deep neural networks for image classification, arXiv preprint arXiv:1202.2745.
4. He, K., G. Gkioxari, P. Dollar and R. Girshick (2018) Mask R-CNN, arXiv preprint arXiv:1703.06870v3.
5. He, K., X. Zhang, S. Ren and J. Sun (2015) Deep residual learning for image recognition, arXiv preprint arXiv:1512.03385v1.
6. He, K., X. Zhang, S. Ren and J. Sun (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, In Proceedings of the IEEE international conference on computer vision, 1026-1034.
7. Ioffe, S. and C. Szegedy (2015) Batch normalization: Accelerating deep network training by reducing internal

covariate shift, arXiv preprint arXiv:1502.03167v3.

8. Kingma, D. P. and J. Ba (2017) Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980v9.
9. Kononenko, I. (1993) Inductive and Bayesian learning in medical diagnosis, *Applied Artificial Intelligence an International Journal*, 7(4), 317-337.
10. Krizhevsky, A., I. Sutskever and G. E. Hinton (2012) ImageNet classification with deep CNNs, Advances in neural information processing systems, 1097-1105.
11. Kumar, P., P. Nagar, C. Arora and A. Gupta (2018) U-SegNet: Fully convolutional neural network based automated brain tissue segmentation tool, arXiv preprint arXiv:1806.04429v1.
12. Le, J. (2018) How to do semantic segmentation using deep learning. Retrieved December 28, 2019, from https://medium.com/nanonets/how-to-do-image-segmentation-using-deep-learning-c673cc5862ef
13. Lin, T. Y., P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongieet (2017) Feature pyramid networks for object detection, arXiv preprint arXiv:1612.03144v2.
14. Long, J., E. Shelhamer and T. Darrell (2015) Fully convolutional networks for semantic segmentation, Proceedings of the IEEE conference on computer vision and pattern recognition, 3431-3440.
15. Nguyen, A., (2019) Scene understanding for autonomous manipulation with deep learning, arXiv preprint arXiv:1903.09761v1.
16. Paszke, A., A. Chaurasia S. Kim and E. Culurciello (2016) Enet: A deep neural network architecture for real-time semantic segmentation, arXiv preprint arXiv:1606.02147v1.
17. Qi, N., K. Voon, M. Ismail and N. Mustaffa (2015) Design and Development of a Mechanism of Robotic Arm for Lifting, 2nd Integrated Design Project Conference (IDPC), Pahang, Malaysia.
18. Ren, S., K. He, R. Girshick and J. Sun (2015) Faster R-CNN: Towards real-time object detection with region proposal networks, Advances in neural information processing systems, 91-99.
19. Ren, S., K. He, R. Girshick and J. Sun (2017) Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149.
20. Ronneberger, O., P. Fischer and T. Brox (2015) U-Net: Convolutional networks for biomedical image segmentation, arXiv preprint arXiv:1505.04597v1.
21. Seetharaman, G., L. Arun and B. Erik (2006) Unmanned vehicles come of age: The DARPA grand challenge, *Computer*, 39(12), 26-29.
22. Simonyan, K. and A. Zisserman (2014) Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556v6.
23. Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich (2015) Going deeper with convolutions, CVPR.
24. Xu, H., Y. Gao, F. Yu and T. Darrell (2017) End-to-end learning of driving models from large-scale video datasets, arXiv preprint arXiv:1612.01079v2.

收件：**108.12.02**　修正：**109.05.04**　接受：**109.07.02**