Pedestrian Segmentation using Efficient Network Architecture

Jiang Chen He
University of Toronto
Toronto, Canada
jiang.he@mail.utoronto.ca

Abstract—**This paper describes an efficient network architecture to segment pedestrians in RGB images. Pedestrian segmentation is useful for many robotic applications, including navigation tasks for self-driving vehicles or human-to-robot interaction tasks for social humanoid robots. With current state-of-the-art segmentation models, they involve complex networks that are resource intensive. This study aims to develop a network with reduced model size and complexity while still producing adequate segmentation quality of pedestrians. The proposed model, termed MUNet, uses a U-Net architecture with a modified MobileNetV2 as the backbone. The results show the proposed network achieves a 75.7% intersection-over-union score for the test dataset with only 0.65M parameters.**

I. INTRODUCTION

Semantic segmentation of images have seen considerable progress over the past decade, paralleled with advancements in deep learning techniques. Current state-of-the-art models can achieve high pixel-level accurate segmentation of image scenery [1] [2] [3]. However, those same models are becoming increasingly complex and demanding in terms of hardware resources. Often the backbones of those models uses ResNet101 [4], which has 44 million parameters, or VGG16 [5], which has 138 million parameters. This can be a barrier for resource constrained applications.

In this paper, we focus on developing a network architecture that can produce quality segmentation while minimizing the size of the model. In particular, we focus on pedestrian segmentation, which is the assignment of labels as pedestrian or background for each pixel in an image. There are many applications to the segmentation of pedestrians in images, from autonomous navigation to surveillance systems to human-robot interaction. This can be especially useful for robots with hardware constraints operating in a pedestrian environment (i.e. garbage picking robot in a park, or a package delivering robot on a university campus). By identifying humans and their physical extents, robots can process this information to avoid or interact with the pedestrians.

The proposed model uses a special type of encoder-decoder architecture called U-Net [6] with a modified MobileNetV2 [7] backbone as the encoder. We experimented with different design variations for the encoder. The performance and complexity of the models with the different modifications are compared against each other and against a fully convolutional network with a ResNet101 and ResNet50 backbone. The model is trained and tested on the CityScape dataset [8].

The remainder of the paper is structured as follows. First, we will review the related works and state of the art. Next, we will describe the datasets used for training and testing and the proposed architectures of the efficient network. Later, we will deliver the results. Finally, concluding remarks and directions for future work will be provided.

II. RELATED WORKS

With the recent surge in deep learning techniques, numerous papers on various semantic segmentation models have been published. Long, Shelhamer, and Darrell presented a fully convolutional network called FCN, which has a large encoder network and a small decoder network [1]. Chen, Papandreou, Kokkinos, Murphy, and Yuille introduced the encoder-decoder network called DeepLabV3, which employs dilated convolutions and uses a spatial pyramid pooling module to capture multiscale information of feature maps [2]. The authors in [3] proposed a pyramid scene parsing network called PSPNet, which uses a pyramid pooling module to separate feature maps into different subregions. These subregions are then upsampled to equal sizes and concatenated. Many other works not mentioned includes [9] [10] [11]. However, these networks require high computational resources and are unsuitable for resource constrained environments.

New approaches to semantic segmentation using efficient networks have been proposed in [12]. This paper introduced the MobileNetV2 architecture and uses it as a feature extractor for DeepLabv3 heads. It achieved 75.32% mIOU with 2.11M parameters. The model was trained on COCO and tested on PASCAL VOC 2012 dataset. Another study developed the encoder-decoder network architecture named SegNet [13]. It uses portions of VGG16 as the encoder with the decoder mirroring the encoder. Max pooling indices at the corresponding encoder layers are recalled during upsampling. The model had 1.425M parameters and achieved 48.5% mIOU with the CamVid dataset.

Architectures similar to the proposed model – U-Net architecture with MobileNetV2 backbone – have been used for fabric defect segmentation [14]. The model had 4.6M parameters and achieved 70% IOU on public and self-built fabric datasets.

Regarding pedestrian segmentation, a study presented a U-Net architecture network that incorporates temporal information [15]. The input comprises of the previous, current, and the following frame to maximize segmentation for the current frame. The network achieved 76.4% mIOU with the CamVid dataset. Brehar, Vancea, Marita, and Nedevschi applied pedestrian segmentation on infrared images using an encoder-decoder network with factorized convolutions [16], achieving 75.4% IOU on their dataset.

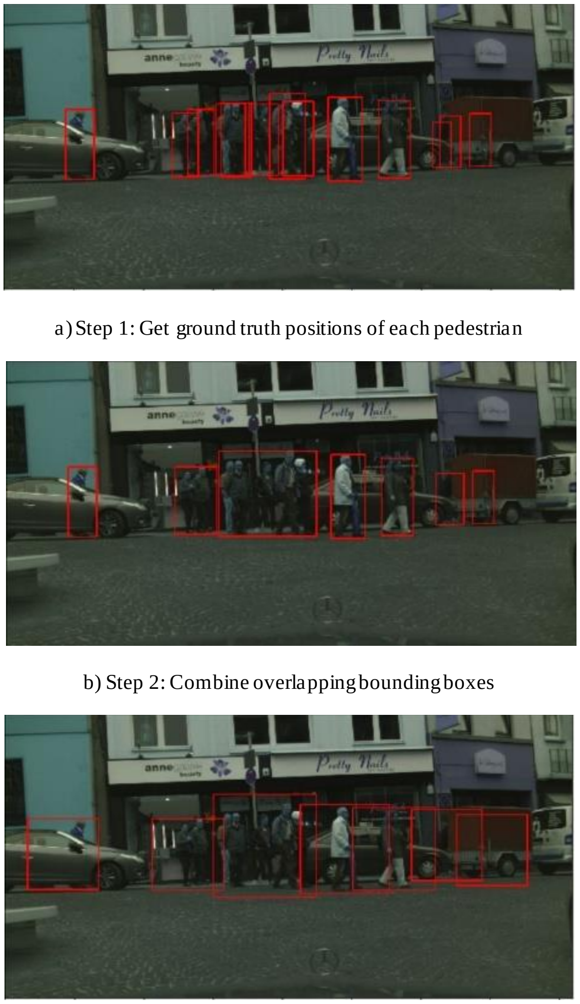
With many of these studies, attempts to achieve high pixel-wise accurate segmentation resulted in networks that tend to be significantly complex. While the studies on semantic segmentation with efficient networks look promising, we aim to develop an even more efficient network architecture for the problem of pedestrian segmentation.

III. Methodology

The pipeline for this project is quite simple, in which an image sample is fed into the proposed network and a binary mask is outputted. This section will describe the methodology for collecting and processing the dataset and constructing the proposed efficient network architecture.

A. *Dataset*

The dataset used in this experiment is the CityScape dataset. It contains 5000 samples of images and high-quality dense pixel annotations. The images were taken in urban environments of 50 different German cities, at various times, seasons, and weather conditions. Since we are only concerned with segmenting pedestrians, samples with no pedestrians were filtered out. By pedestrians, we also refer to cyclists and other riders. That left us with 2175 samples.



a) Step 1: Get ground truth positions of each pedestrian

b) Step 2: Combine overlapping bounding boxes

c) Step 3: Resize each bounding box to a square

Fig 1. Preprocessing steps for a CityScape dataset sample image

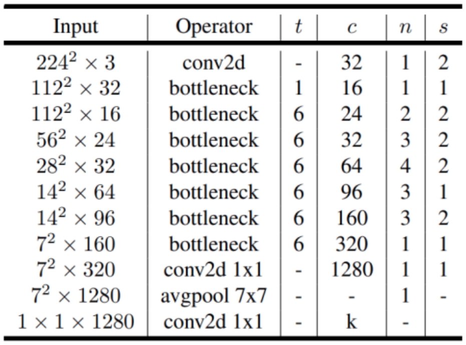
Each sample in the dataset were of size 1024 x 2048. However, the proposed model is designed to take in an image with a size of 256 x 256. Aggressively scaling down the image and ground truth segmentation mask was impractical due to significant corruption in annotation quality. To address this concern, we decided to use ground truth pedestrian bounding boxes as regions of interest in the images, and then shape them into new samples. First, the ground truth bounding boxes were extracted to get the positions of each pedestrian. Next, the bounding boxes with any overlap were combined together. Finally, the remaining bounding boxes were resized to a square, with minimum size of 256 x 256, with the pedestrians randomly off centered in the bounding boxes. For each bounding box, the image region along with the corresponding region in the segmentation mask became new samples for the dataset. An example of the preprocessing procedure for a sample image is shown in Figure 1.

In total 5518 new samples were created. Pedestrians with bounding box heights smaller than 100 pixels were filtered out, as annotation quality became questionable for those pedestrians. Samples with size greater than 256 x 256 were scaled down to the desired size.

B. *Proposed Network Architecture*

The proposed model, termed MUNet, uses the U-Net architecture with a modified MobileNetV2 as the feature extractor. It is a memory-efficient network for pedestrian segmentation.

MobileNetV2 is a light-weight model made by Google tailored for mobile devices. It has 2.2 million parameters, considerably less than ResNet or VGG. Figure 2 shows the overall architecture of MobileNetV2. It has 16 bottlenecks, where each bottleneck consists of a 1x1 convolutional layer, 3x3 depth-wise convolutional layer, followed by a 1x1 convolutional layer with a linear activation function. The bottleneck with a stride of 1 has skip connection at the input and output of the block. It is referred to as an inverted residual block. While classical residuals connect layers with high number of channels, inverted residuals connect layers with low number of channels. The structure of a bottleneck block is given in Figure 3.



Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Fig 2. MobileNetV2 architecture; t - expansion factor, c - the output channel, n – block repeating number, s – stride.
Source: [12, Table. 2.]

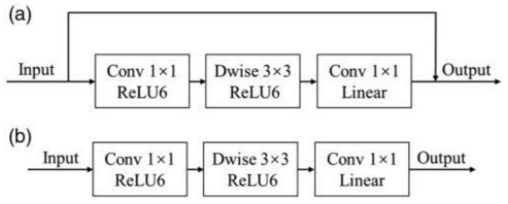


Fig 3. Bottleneck structure with a) stride = 1, b) stride = 2.
Source: [14, Fig 4.]

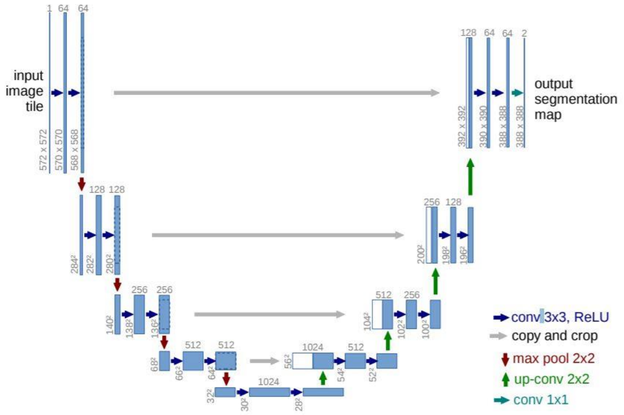


Fig 4. U-Net architecture
Source: [6, Fig. 1.]

U-Net is an encoder-decoder network in which features maps from the encoder path are concatenated with the corresponding feature maps from the decoder path during upsampling. It was originally used to segment medical images of tissues. The U-Net architecture is illustrated in Figure 4. For our proposed model, only the feature maps from the input layer, 1st bottleneck, 3rd bottleneck, and 6th bottleneck are concatenated. The full decoder network is shown in Table 1. Each convolution output in the decoder is batch normalized and applied a ReLu activation function. The final convolution output is applied a sigmoid activation function.

IV. Experimental Results

We experimented with different modification to MobileNetV2. We compare the performance of the network when MobileNetV2 prematurely outputs at the 7th, 10th, 13th, and 16th bottlenecks, in which the network will be referred to as MUNet7, MUNet10, MUNet13, and MUNet16, respectively. To evaluate their efficiency as a network, they are also compared against other deep learning-based segmentation models: FCN with a ResNet101 backbone and FCN with a ResNet50 backbone.

A. *Model Training*

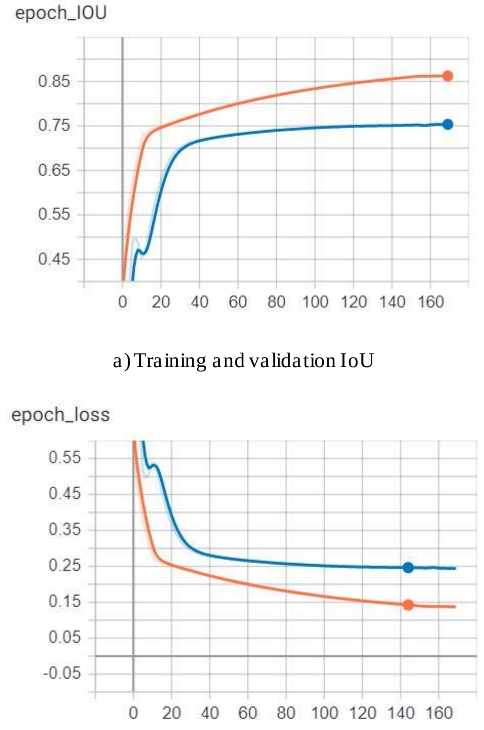
The models were trained and evaluated on the Google Colab environment with a GPU enabled. The deep learning framework Tensorflow was used to build the networks. The dataset of 5518 samples were split into 80% training set, 10% validation set, and 10% testing set. For each experiment, the model was trained with an Adam optimizer with batch size of 32 and an initial learning rate of 0.001 for 200 epochs. We incorporated learning rate scheduler and early stopping to prevent overfitting. A summary of the hyperparameters are provided in Table II.

TABLE I
Decoder Network Parameter

Input	Operator
-	Conv2DTranspose 2x2
32^2 x 384	Concatenate with output of Block 6
32^2 x 576	Conv2D 3x3
32^2 x 64	Conv2DTranspose 2x2
64^2 x 64	Concatenate with output of Block 3
64^2 x 208	Conv2D 3x3
64^2 x 48	Conv2DTranspose 2x2
128^2 x 48	Concatenate with output of Block 1
128^2 x 144	Conv2D 3x3
128^2 x 32	Conv2DTranspose 2x2
256^2 x 32	Concatenate with output of input layer
256^2 x 35	Conv2D 3x3
256^2 x 16	Conv2D 1x1

TABLE II
HyperParameter Summary

Hyperparameter	Value
Test Split	80-10-10
Epochs	200
Batch Size	32
Learning Rate (LR)	0.001
LR Scheduler Factor	0.1
LR Scheduler Patience	4



a) Training and validation IoU

b) Training and validation loss

Fig 5. IoU and loss during training. Orange represents training and blue represents validation.

The images and segmentation masks are preprocessed before training. The images are normalized and downsized to size of 256 x 256. The segmentation masks are also downsized and all labels except for the pedestrian label are removed from the mask.

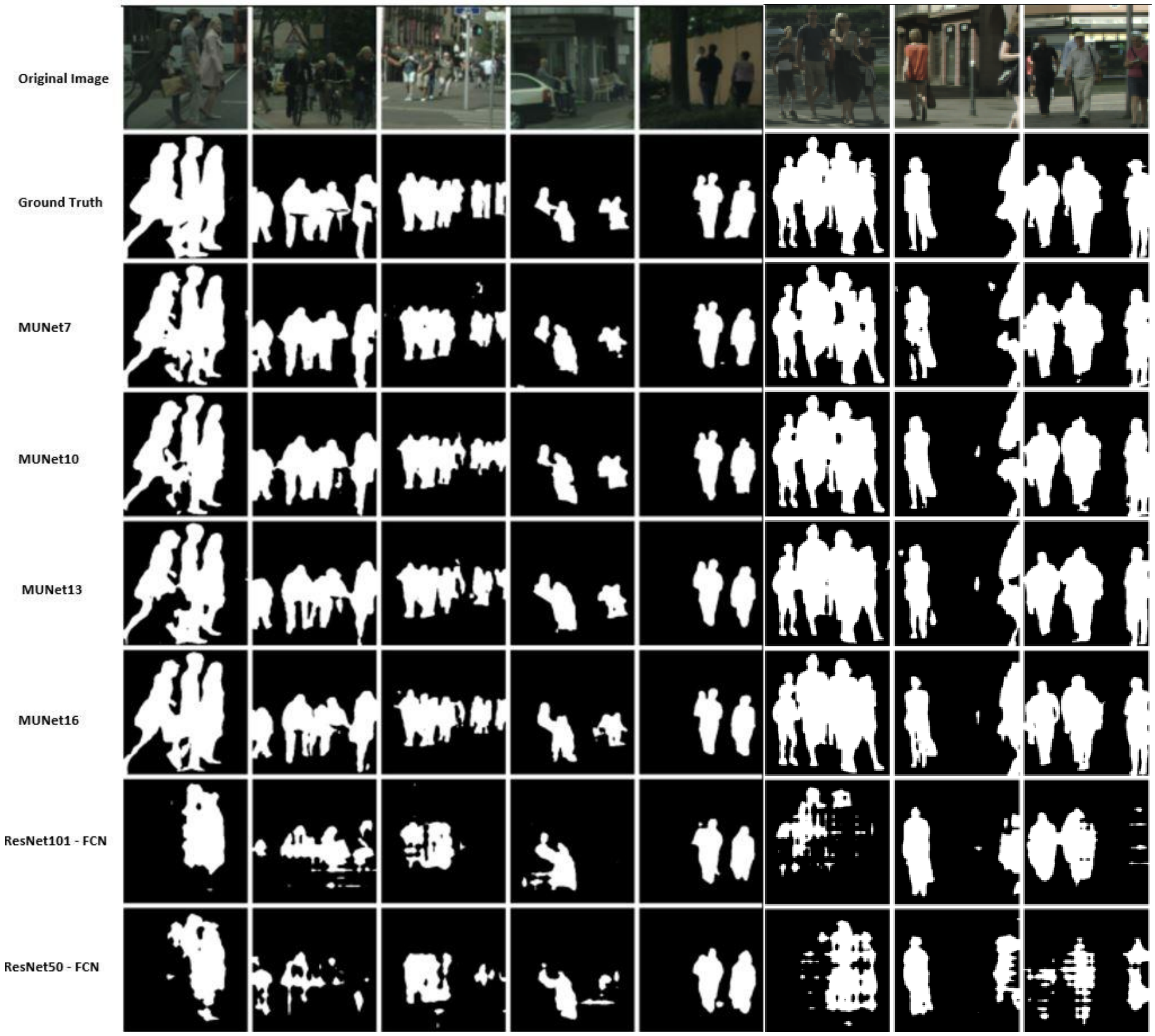


Fig 6. Qualitative results for 8 test samples

The metric used to evaluate the performance of the model is the intersection over union (IoU). IoU is described as

$$IoU = \frac{TP}{FN + TP + FP} \quad (1)$$

where TP is the true positive, FN is the false negative, and FP is the false positive. True positive represents the number of pixels correctly assigned the pedestrian label, false negative represents the number of pixels incorrectly assigned the background label, and false positive represents the number of pixels incorrectly assigned the pedestrian label. It was found that a loss function of

$$L = 1 - IoU \quad (2)$$

provided the best results. An example of training results for a network, MUNet7, is shown in Figure 5.

B. Results and Analysis

The performance and efficiency of the modified networks were evaluated against each other and against pretrained FCNs with ResNet101 backbone and ResNet50 backbone provided by the deep learning framework PyTorch. It is worth noting that the FCNs have not been trained on our CityScape dataset, thus they are expected to perform worse as the MUNets have an unfair advantage. The qualitative results on test set can be seen in Figure 6. The quality of segmentation from each MUNet does not differ significantly and have accurate and detailed structures resembling the ground truth segmentation. The FCNs produced inaccurate segmentations, as expected.

A comparison of the models is presented in Table III. Regarding the size of the models, the MUNets are significantly smaller than the FCNs; the smallest MUNet7 have only 0.65M parameters. Comparing MUNet7 to MUNet16, model size is reduced by almost 80% and inference time is reduced by almost 50% without a large

TABLE III
MODEL EFFICIENCY COMPARISON

	Number of Parameters	**Model Size**	**Mean Inference Time**	**Test mIoU**
MUNet7	**0.65M**	**7.6MB**	**23.8ms**	**75.7%**
MUNet10	0.81M	9.5MB	31.2ms	76.6%
MUNet13	1.26M	14.8MB	37.2ms	78.7%
MUNet16	3.05M	35.6MB	44.6ms	79.5%
ResNet101-FCN	54.31M	208MB	34.4ms	52.4%
ResNet50-FCN	35.32M	136MB	20.1ms	52.6%

drop on the mean test IoU score. The performance of MUNet7 is verified by the qualitative results, with MUNet7 still delivering accurate segmentation masks even with reduced number of parameters. Thus MUNet7 is selected as the final proposed model. Unfortunately, computational efficiency is poor, with MUNet7 having slightly greater inference time than ResNet50–FCN, even though the proposed network is 18 times smaller. This can be an area of future investigation.

Overall, these results show that a memory-efficient network with only 0.65M parameters is able to perform accurate pedestrian segmentation, with inference time slightly worse or comparable to state-of-the-art segmentation networks. Extensive hyperparameter tuning as well as generating additional training data can be used to further improve the proposed network performance.

V. DISCUSSION

A limitation to this approach is that by using the ground truth bounding boxes to make new samples using the procedure outlined in Section III, the training data become mostly composed of samples with pedestrians vertically centered and taking up most of the image. This may negatively affect the network's scale-invariance. One way to solve this problem is to randomize the final resizing of the bounding boxes to include bigger squares. Another way is to introduce data augmentation by having multiple resizing for the same bounding box.

Due to computational limitations of Google Colab, the FCNs were not retrained on the CityScape dataset. Similar reasons pertain to other state-of-the-art deep learning-based segmentation models. Further comparisons with other network architectures, such as SegNet, trained on our CityScape dataset would have provided a deeper understanding of the model results.

The proposed network performs the best when only a few upstanding pedestrians are in the image. It is also able to segment occluded pedestrians fairly accurately. A challenging task for the network is determining when to include accessories such as purses and backpacks as part of the pedestrian segmentation. It also often fails when clothing have similar color to the background. For example if a pedestrian wearing black pants stands in front of a black car, the black pants may be identified as part of the car, which will be labeled as the background.

The proposed network has a slightly inferior inference time compared to the ResNet50-FCN. A factor that may have contributed to this discrepancy is the different deep learning frameworks used for evaluating the MUNets and FCNs. The MUNets were built using Tensorflow, while the pretrained FCNs were provided by PyTorch. Evaluating all the models on one deep learning framework may entail better comparison results.

Directions for future research can be to experiment with the network architecture, such as employing a different upsampling approach, to improve the inference runtime and network performance. For example, instead of concatenating the feature maps of the encoder layer to the corresponding decoder layer, one may try using the max pooling indices of the encoder layer for deconvolution during upsampling, inspired by the SegNet architecture. Another extension to this study can be to investigate different loss functions for improving network performance.

VI. CONCLUSION

This paper proposed MUNet, a memory-efficient network architecture that shows promising results for pedestrian segmentation. The network incorporated the U-Net architecture with a MobileNetV2 backbone. Using the MobileNetV2 up to the 7th bottleneck greatly reduced the model size while maintaining quality segmentation results. Its performance was compared with a fully convolutional network with ResNet backbones. It achieved greater memory efficiency but slightly inferior runtime. Areas for improvement includes tweaking the model architecture to decrease network inference time and testing the proposed network on more diverse datasets.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015.

[2] 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 Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, Apr. 2018.

[3] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, doi: 10.1109/cvpr.2016.90.

[5] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," presented at the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Nov. 2015.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Lecture Notes

in Computer Science, Springer International Publishing, 2015, pp. 234–241.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” presented at the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2018.

[8] M. Cordts et al., “The Cityscapes Dataset for Semantic Urban Scene Understanding,” presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016.

[9] H. Noh, S. Hong, and B. Han, “Learning Deconvolution Network for Semantic Segmentation,” presented at the 2015 IEEE International Conference on Computer Vision (ICCV), Dec. 2015.

[10] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, “The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation,” presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jul. 2017.

[11] W. Liu, A. Rabinovich, and A. C. Berg, “Parsenet: Looking wider to see better,” CoRR, vol. abs/1506.04579, 2015

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” presented at the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2018.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” IEEE Trans. Pattern Anal. 2495, Dec. 2017.

[14] J. Jing, Z. Wang, M. Rätsch, and H. Zhang, “Mobile-Unet: An efficient convolutional neural network for fabric defect detection,” Textile Research Journal, p. 004051752092860, May 2020.

[15] M. Ullah, A. Mohammed, and F. Alaya Cheikh, “PedNet: A Spatio-Temporal Deep Convolutional Neural Network for Pedestrian Segmentation,” J. Imaging, vol. 4, no. 9, p. 107, Sep. 2018.

[16] R. Brehar, F. Vancea, T. Marita, and S. Nedevschi, “A Deep Learning Approach For Pedestrian Segmentation In Infrared Images,” presented at the 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), Sep. 2018.