Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8684138, 12 pages
https://doi.org/10.1155/2022/8684138

Research Article

A Deep Learning-Based Semantic Segmentation Architecture for Autonomous Driving Applications

Sharjeel Masood[1], **Fawad Ahmed**[2], **Suliman A. Alsuhibany**[3], **Yazeed Yasin Ghadi**[4], **M. Y. Siyal**[5], **Harish Kumar**[6], **Khyber Khan**[7] and **Jawad Ahmad**[8]

[1]*Healthhub, Seoul, Republic of Korea*
[2]*Department of Cyber Security, Pakistan Navy Engineering College, NUST, Pakistan*
[3]*Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia*
[4]*Department of Computer Science and Software Engineering, Al Ain University, Abu Dhabi 122612, UAE*
[5]*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*
[6]*Department of Computer Science, College of Computer Science, King Khalid University, Abha 61413, Saudi Arabia*
[7]*Department of Computer Science, Khurasan University, Jalalabad, Afghanistan*
[8]*School of Computing, Edinburgh Napier University EH10 5DT, UK*

Correspondence should be addressed to Khyber Khan; khyber.khan.khurasan@gmail.com

Received 30 March 2022; Revised 21 May 2022; Accepted 2 June 2022; Published 18 June 2022

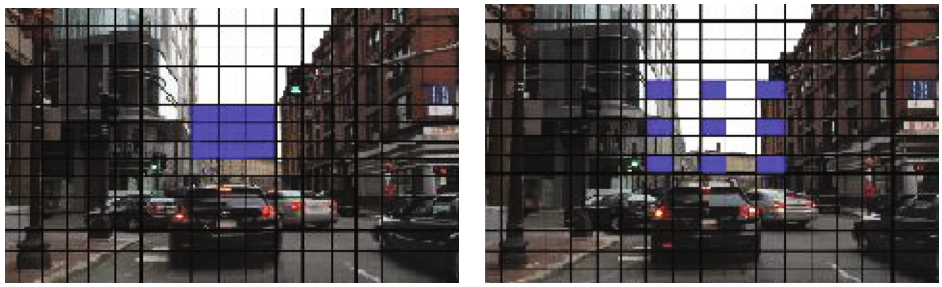
Academic Editor: Farhan Ullah

In recent years, the development of smart transportation has accelerated research on semantic segmentation as it is one of the most important problems in this area. A large receptive field has always been the center of focus when designing convolutional neural networks for semantic segmentation. A majority of recent techniques have used maxpooling to increase the receptive field of a network at an expense of decreasing its spatial resolution. Although this idea has shown improved results in object detection applications, however, when it comes to semantic segmentation, a high spatial resolution also needs to be considered. To address this issue, a new deep learning model, the M-Net is proposed in this paper which satisfies both high spatial resolution and a large enough receptive field while keeping the size of the model to a minimum. The proposed network is based on an encoder-decoder architecture. The encoder uses atrous convolution to encode the features at full resolution, and instead of using heavy transposed convolution, the decoder consists of a multipath feature extraction module that can extract multiscale context information from the encoded features. The experimental results reported in the paper demonstrate the viability of the proposed scheme.

1. Introduction

Computer vision stands as the backbone of various modern autonomous driving systems [1] with semantic segmentation being one of its fundamental tasks. The goal of semantic segmentation is to assign a label to every pixel of an image. Deep convolutional neural networks have opened up a wide area of extremely effective solutions to problems like object detection [2], lane detection [3], object tracking [4], and semantic segmentation.

Improvements in the performance of deep neural networks have largely been achieved by increasing the number of learnable parameters along with careful network designing, making them computationally expensive. Reducing computational cost and extracting the maximum possible performance from the minimum number of learnable parameters is undoubtedly an extremely important requirement when dealing with embedded systems in autonomous driving. To detect large objects in an image, it is necessary to have a receptive field large enough to gather enough context information, and the use of pooling layers in many recent networks to increase the receptive field means that this information is found on a courser scale at higher layers. Finer details like edges of an object or small/thin objects



(a) Dilation = 1 (b) Dilation = 2

FIGURE 1: The pixels in the blue contribute in the calculation of the center pixel in the output feature map. (a) shows a 3×3 convolution with dilation rate of 1. (b) shows a 3×3 convolution with dilation rate of 2.

need high spatial resolution to perform accurate segmentation.

To increase the receptive field of the network, the encoder in encoder-decoder methods normally downsamples the image using strided convolution or pooling layers or both, at an expense of reducing the spatial resolution. The decoder then uses transposed convolution to upsample these encoded features to obtain a high-resolution final feature map; this makes segmenting small objects difficult. Encoder-decoder structures like FCN [5] and U-Net [6] use skip connections to connect the lower layers in an encoder to higher layers of the decoder; this partially solves the problem by allowing both high layer course information and low layer fine information to contribute to the prediction of the final feature map. This technique is effective to some extent but can lead to deeper models with a large number of learnable parameters.

An alternative way can be to maintain the spatial resolution of the features in the encoder while using atrous convolution to increase the receptive field. DeepLab [7] modifies FCN [5] by replacing the last 2 downsampling operations with atrous convolutions to maintain the receptive field. In the architecture proposed by [8], atrous convolutions are used extensively to effectively increase the receptive field while maintaining the spatial resolution throughout the network to segment smaller objects. Figure 1 shows how atrous convolution expands the receptive field by adding holes into a normal convolutional layer. A convolution layer with a 3×3 kernel and a dilation rate of 2 has the same field of view as a layer with a 5×5 kernel, while only using 9 parameters. Dilated convolution is an effective way to maintain spatial resolution, but going deeper with high-resolution feature maps can also introduce latency in the system. Processing features in full resolution can be computationaly expensive, to reduce the latency in our system we used maxpooling half way down our network to reduce the spacial resolution by half, this reduces the run time of our network and at the same time increases the receptive field for larger objects. Capturing useful image context information at multiple scales has proven to enhance segmentation accuracy. Pyramid pooling modules like the one introduced in [9] uses pyramid pooling operation for multiple scale context aggregation. The authors in [10] divided the initial input into four subregions and obtained the pooled features from each of those four subregions, respectively. DeepLab [7] on the other hand use atrous spatial pyramid pooling(ASPP) that exploits atrous convolution to divide the features into different scales instead of pooling layers. A deeper version of the ASPP module was introduced in [11] by adding a standard 3×3 convolution after 3×3 atrous convolutions. We have taken a similar approach by using a multipath feature extraction module as a decoder to fuse together the key information from three different scales, leading to better segmentation ability.

2. Related Work

Semantic segmentation is of great importance in self-driving cars and various driving aids. Deep convolutional neural networks when used in encoder-decoder network architectures have shown remarkable segmentation performance. Encoder-decoder network architectures were first introduced by Bayesian SegNet [12] and SegNet [13]; they used the encoder to downsample the features, and then, the decoder was responsible for recovering the spacial dimensions of the features. FCN [5] used a similar approach by using a classification model like VGG [14] as an encoder to extract features and those extracted features were then upsampled to perform pixelwise prediction in full resolution.

Recent works have brought various changes to the encoder-decoder structure. Instead of using transposed convolution in the decoder, the architecture in [15] introduced a JPU unit to decode the features encoded by FCN [5], the joint pyramid upsampling (JPU) unit upsamples the last 3 feature maps from FCN and then uses 4 dilated convolution layers to extract the features at multiple scales; this decreases the size of the network and also speeding up the network. Encoder-decoders like the ones in [16, 17] use an encoder to extract multilevel features and then used a decoder to combine them into a high-resolution final prediction, avoiding the extensive use of transposed convolution.

DeepLabs [7, 18] introduced atrous spatial pyramid pooling (ASPP) to extract context information at different scales for better segmentation. PSPNet [9] used global average pooling to capture context information. A similar multipath module has been used by [19] to generate a feature

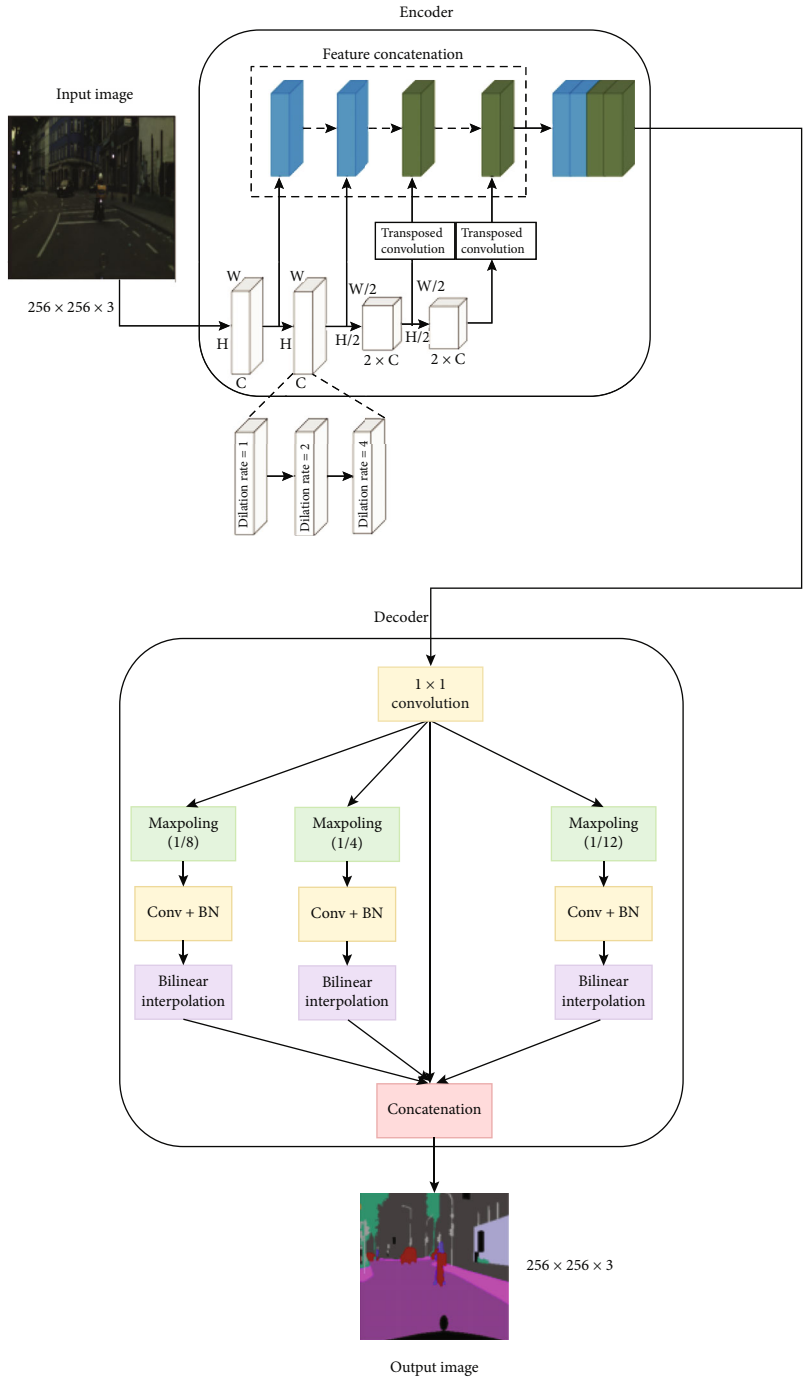


Figure 2: Structure of architecture 1: M-Net encoder+PSP decoder.

pyramid in a generative adversarial network for road segmentation. The authors in [20] use multiple paths in the decoder to capture different variations in the face with the same expression label. In [21], the input is taken at three different scales and an attention map for each scale is then learned. Yap [22] proposed an architecture to segment damages on the road; the architecture contained detail branch and segmentation branch using VGG net [14] and MobileNetV2 [23], respectively, as backbone architectures.

All these developments lead to a huge improvement in prediction accuracy but some of them are hard on computations. There have been developments to reduce the computational complexity required to achieve certain segmentation accuracy. ENet [24] used early downsampling to reduce the cost of processing large frames and used PReLU as activation. The use of PReLU tends to increase the computational cost, but the reduction in computations caused by reducing the spatial dimensions of the features early in the network was large enough to make the overall network faster than its counterparts. SINet [25] introduced an extremely lightweight multipath structure containing spacial squeeze modules. These spacial squeeze modules

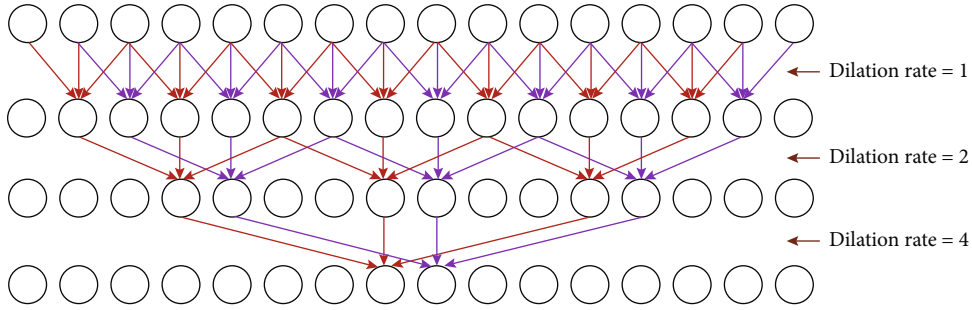


Figure 3: Connections between 3 convolutional layers of dilation factors 1, 2, and 4.

Table 1: Detailed architecture of the encoder.

	Layers	Input	Output	
	Conv2d	$32 \times 56 \times 256$	$32 \times 256 \times 256$	$k = 3, p = 1, d = 1$
Conv-block 1	Conv2d	$32 \times 256 \times 256$	$32 \times 256 \times 256$	$k = 3, p = 2, d = 2$
	Conv2d	$32 \times 256 \times 256$	$32 \times 256 \times 256$	$k = 3, p = 4, d = 4$
	Conv2d	$32 \times 256 \times 256$	$32 \times 256 \times 256$	$k = 3, p = 1, d = 1$
Conv-block 2	Conv2d	$32 \times 256 \times 256$	$32 \times 256 \times 256$	$k = 3, p = 2, d = 2$
	Conv2d	$32 \times 256 \times 256$	$32 \times 256 \times 256$	$k = 3, p = 4, d = 4$
	MaxPooling	$32 \times 256 \times 256$	$32 \times 128 \times 128$	$k = 3, s = 2, p = 1$
	Conv2d	$32 \times 128 \times 128$	$64 \times 128 \times 128$	$k = 3, p = 1, d = 1$
Conv-block 3	Conv2d	$64 \times 128 \times 128$	$64 \times 128 \times 128$	$k = 3, p = 2, d = 2$
	Conv2d	$64 \times 128 \times 128$	$64 \times 128 \times 128$	$k = 3, p = 4, d = 4$
	Conv2d	$64 \times 128 \times 128$	$64 \times 128 \times 128$	$k = 3, p = 1, d = 1$
Conv-block 4	Conv2d	$64 \times 128 \times 128$	$64 \times 128 \times 128$	$k = 3, p = 2, d = 2$
	Conv2d	$64 \times 128 \times 128$	$64 \times 128 \times 128$	$k = 3, p = 4, d = 4$

k denotes the kernel size, p denotes the padding used, d denotes the dilation rate, and s denotes the strides.

reduce the number of feature maps by half by using pointwise convolution, to further reduce the computations they used average pooling to squeeze the resolution of the feature maps, beating ENet [24] in the total number of parameters.

3. Proposed Method

This section will discuss our proposed methodology in detail. Our encoder is designed to effectively encode the features in full resolution without allowing too much latency into the system. Since our encoded features will be in full resolution it would eliminate the need to use extensive transposed convolutions in our decoder. The decoder in our case is a multipath feature extraction module; this would extract features at different scales, making better use of high-resolution encoded features. We have proposed two architectures both with the same encoder but one with PSP module as the decoder and the second one with ASPP module as the decoder.

3.1. Architecture 1: M-Net Encoder+PSP Decoder. The encoder is aimed at encoding the features at full resolution making much finer predictions possible, while also having a large enough receptive field to effectively segment large objects.

Our encoder is four conv-blocks deep as shown in Figure 2. Each conv-block has one standard convolutional layer and two atrous convolutional layers with dilation rates of 2 and 4, respectively, and each of them with a 3×3 kernel. Stacking up convolutional layers in this particular order connects each output pixel with 15×15 input pixels. To explain this concept, we have used 1D convolutions to make things look a bit less complicated. Figure 3 shows a set of 1D convolutions each with a kernel size of 3 and a dilation factor of 1, 2, and 4 is used for convolutional layers going from the top, middle, to bottom layers, respectively. Each conv-block effectively increases the receptive field by 15 pixels while maintaining constant spatial dimensions; this order of dilation rate also avoids the problem where the information from the adjacent pixels do not overlap if only even dilation rates are used as pointed out by [8]. Since going deep with high spatial resolution can be computationally expensive, the first 2 conv-blocks are followed by a MaxPooling layer which reduces the spatial dimensions of the features by half, after which 2 more conv-blocks are added. This also helps to increase the receptive field of the network and enables it to segment larger objects in the image. Table 1 shows the input and output dimensions of every layer. We selected a kernel size of 3 for each layer throughout the

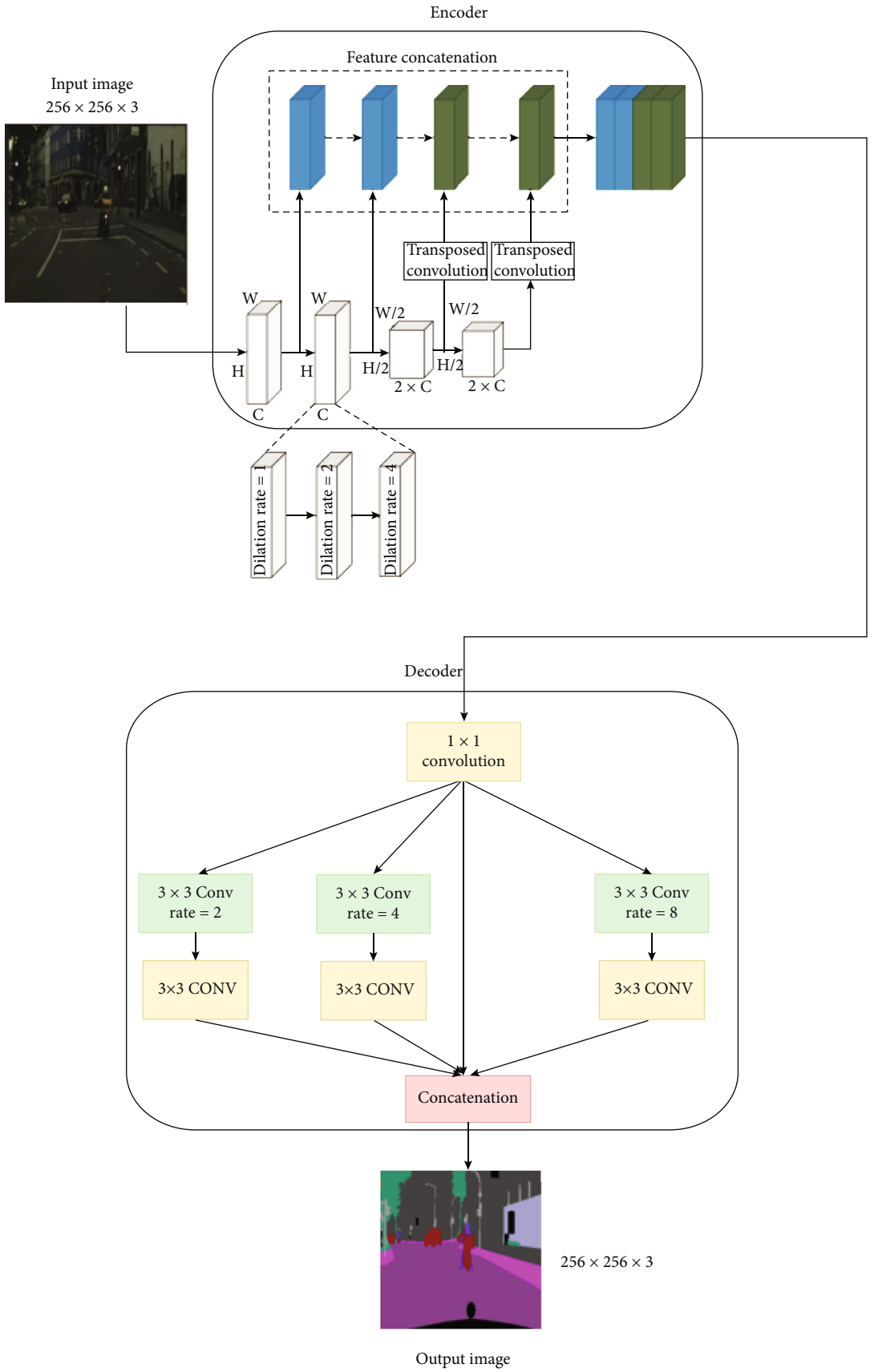


Figure 4: Structure of architecture 2: M-Net encoder+ASPP decoder.

Table 2: The architectural difference between PSP and ASPP modules.

	PSP		ASPP	
Branch	Layers	Parameters	Layers	Parameters
Branch 1	MaxPooling	$k = 9, s = 8, p = 1$	Conv2d	$k = 3, p = 8, d = 8$
	Conv2d	$k = 3, s = 1, p = 1$	Conv2d	$k = 3, p = 1, d = 1$
Branch 2	MaxPooling	$k = 5, s = 4, p = 1$	Conv2d	$k = 3, p = 4, d = 4$
	Conv2d	$k = 3, s = 1, p = 1$	Conv2d	$k = 3, p = 1, d = 1$
Branch 3	MaxPooling	$k = 3, s = 4, p = 1$	Conv2d	$k = 3, p = 2, d = 2$
	Conv2d	$k = 3, s = 1, p = 1$	Conv2d	$k = 3, p = 1, d = 1$

Table 3: Experimental results of segmentation models on Cityscapes.

Model	Number of parameters	Size	mIoU
ENet	688 K	3 Mb	53
M-Net+ASPP	375 K	1.5 Mb	58
M-Net+PSP	348 K	1.38 Mb	56
SINet	43 K	0.3 Mb	43

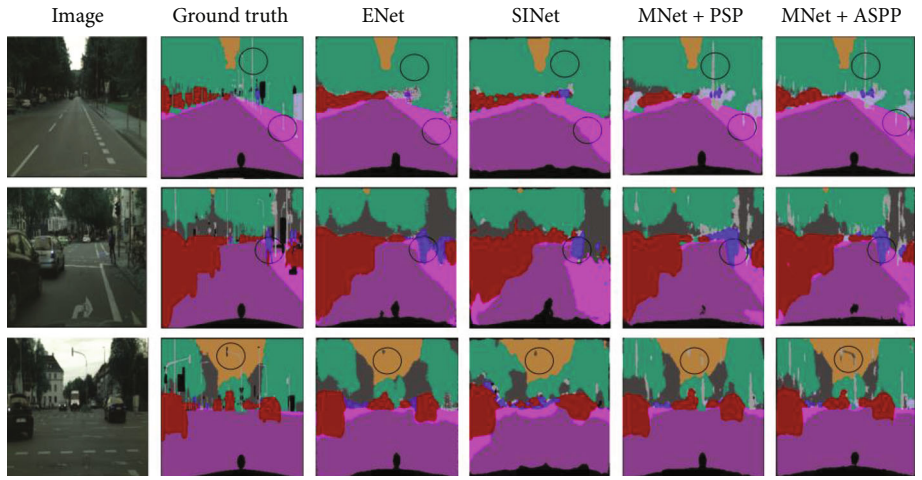


Figure 5: Visual comparison of segmentation masks with different models on Cityscapes dataset.

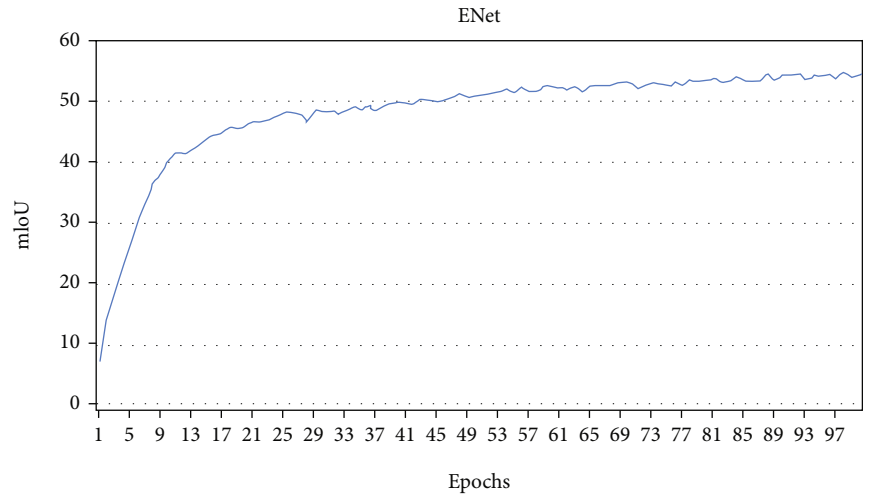
network; we have avoided using larger kernel sizes to reduce computations. Specific padding is used for each dilation rate to maintain the spacial resolution. The outputs from the last two conv-blocks are upscaled using transposed convolution to recover the spatial dimensions of the features from the last 2 conv-blocks. All four feature maps are then concatenated together resulting in a feature map of shape $128 \times 256 \times 256$ which is then passed on to the decoder which in this first case is a PSP module.

The emphasis behind using a PSP module as the decoder is to extract features from different scales further increasing the receptive field and to fuse the information received from different scales. This increases the range of context information obtained.

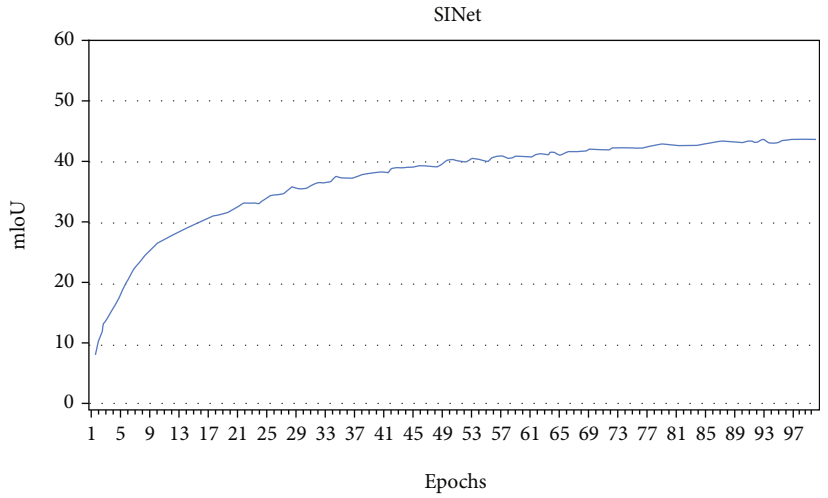
This idea was inspired by the PSP module proposed by [9] which uses spacial pyramid pooling to capture the global context information from the high-resolution features.

Multipath structures like the ones used in google's inception nets and the ones used in this PSP module can be hard on computations. To counter the high computational requirements, we have used a 1×1 convolution layer to reduce the number of channels. The feature maps are then pooled into their respective subregions each followed by a 3×3 convolution layer and batch normalization as shown in Figure 2. The features from each scale are then upsampled using bilinear interpolation and are then concatenated together.

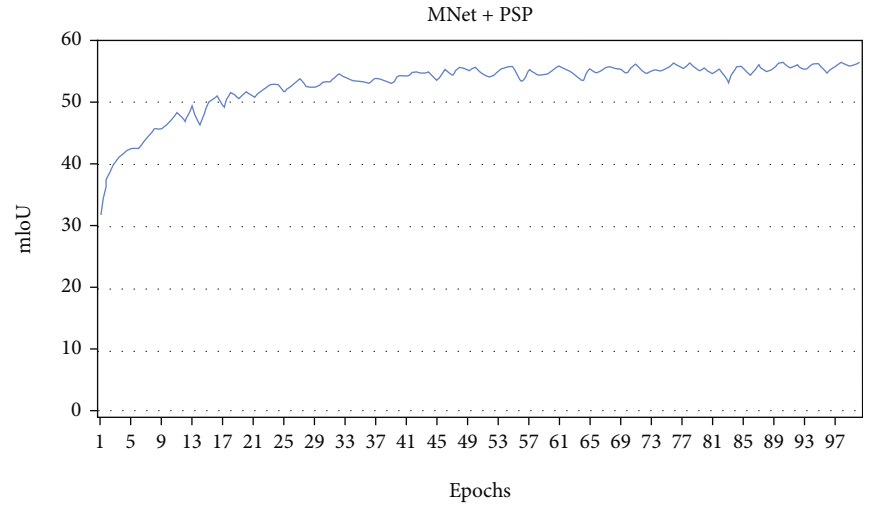
3.2. Architecture 2: M-Net Encoder+ASPP Decoder. Another way to extract multiple-scale information is by using atrous spatial pyramid pooling (ASPP). The ASPP module replaces pooling layers with atrous convolution at different dilation rates to extract features at multiple scales. The reason why we have not completely gone with pooling layers in the PSP module to extract multiscale features is that despite



(a)

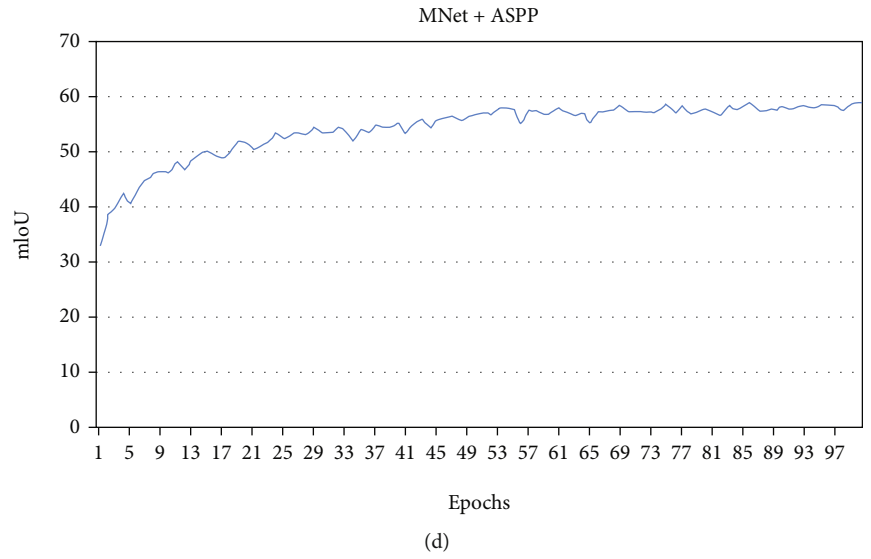


(b)



(c)

Figure 6: Continued.



(d)

Figure 6: Graphical comparison of validation mIoU against epoch on the Cityscapes dataset. (a) The graph of ENet. (b) The graph of SINet. (c) The graph of M-Net with a PSP decoder. (d) The graph of M-Net with ASPP decoder.

Table 4: Experimental results of segmentation models on Mapillary Vistas.

Model	Number of parameters	Size	mIoU
ENet	688 K	3 Mb	56
M-Net+ASPP	375 K	1.5 Mb	61
M-Net+PSP	348 K	1.38 Mb	59
SINet	43 K	0.3 Mb	50

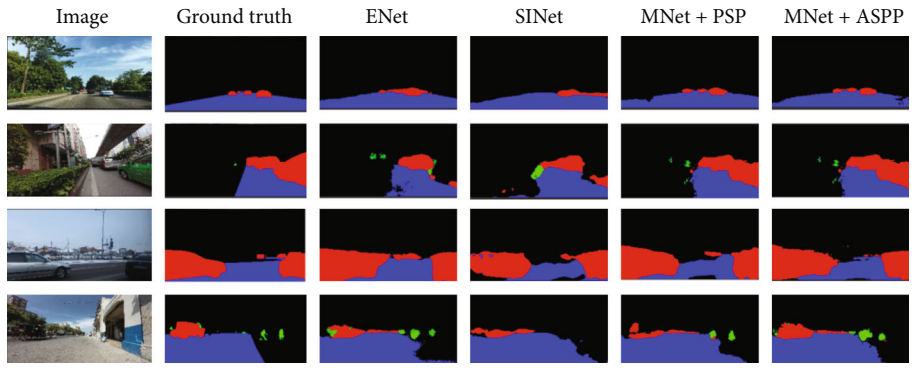
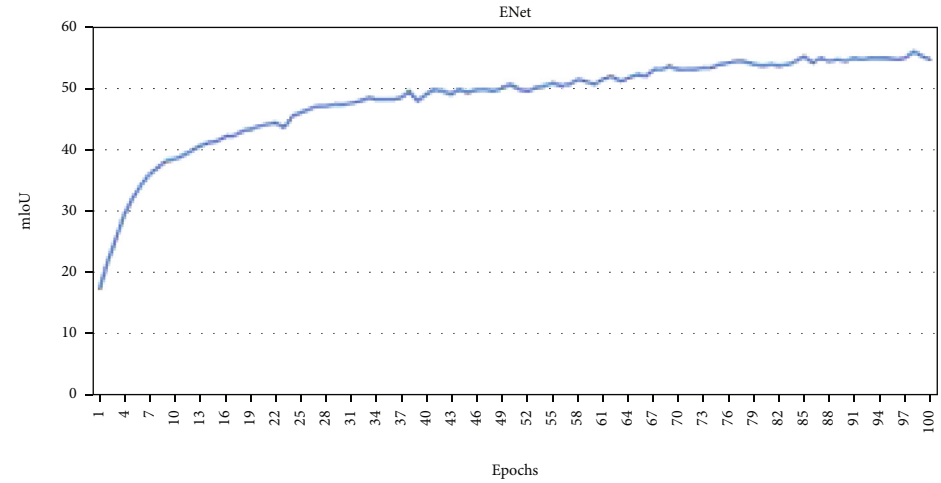


Figure 7: Visual comparison of segmentation masks with different models on Mapillary Vistas dataset.

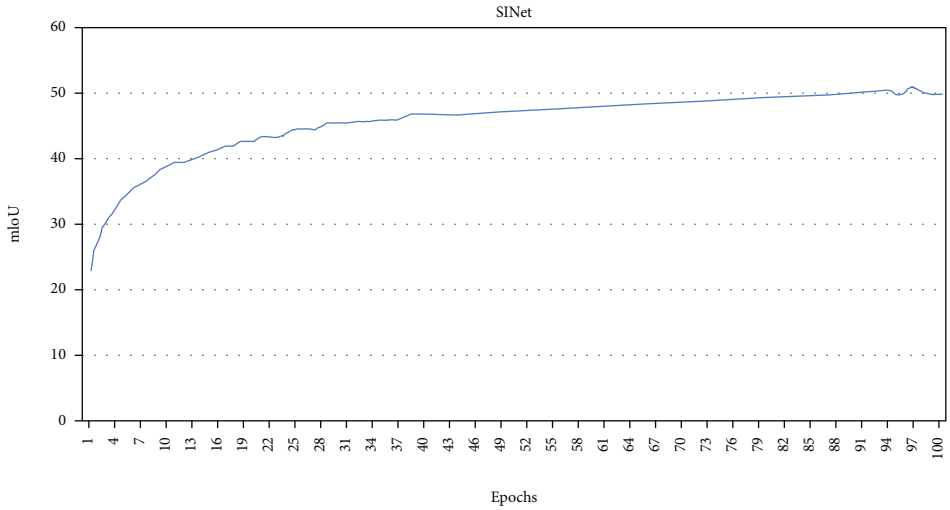
being robust at increasing the receptive field of the network maxpooling layers have shown to lose some of the information; this effect is shown by the authors in [26], and we have also observed finer results with ASPP module. We have used three atrous convolution layers with the dilation rates of 2, 4, and 8, respectively. Each atrous convolution is followed by a standard convolution layer with a 3×3 kernel as shown in Figure 4. We decided not to go deep with the convolution layers in PSP and ASPP modules as a large number of computations on multiple paths can make the system slower. Table 2 shows the architectural difference between our PSP and ASPP decoders.

4. Experiments and Results

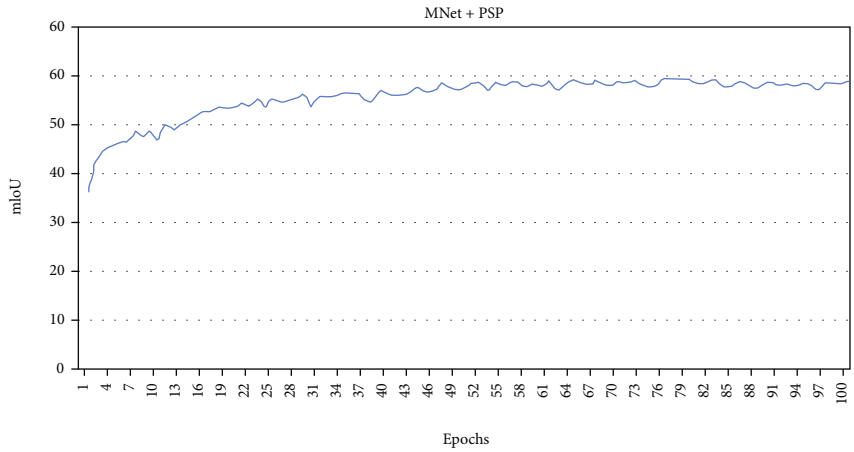
We have used Pytorch as our deep learning framework to train and test our model. Adam Optimizer [27] with a learning rate of $4e-6$, weight decay of $2e-4$, and batch size of 10



(a)

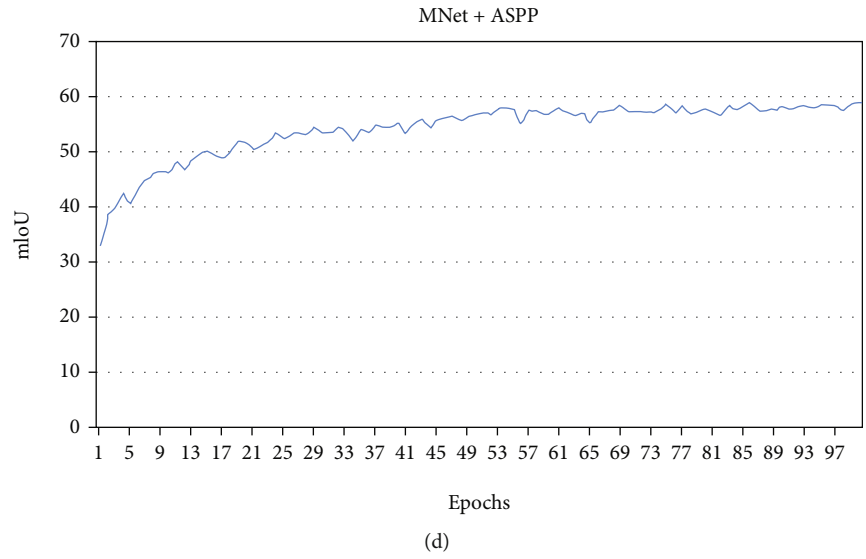


(b)



(c)

Figure 8: Continued.



(d)

Figure 8: Graphical comparison validation mIoU against epoch on the Mapillary dataset. (a) The graph of ENet. (b) The graph of SINet. (c) The graph of M-Net with a PSP decoder. (d) The graph of M-Net with ASPP decoder.

was used to train our networks on Cityscapes [28] and Mapillary vistas [29]. We have compared our results with ENet and SINet since they both are known for working with a low number of parameters.

We have used mean intersection over union as our evaluation metric; mIoU is the mean of IoU scores for each class Equation (1), where TP, FP, and FN represent true positives, false positives, and false negatives, respectively.

$$\text{mIoU} = \frac{1}{C}\sum_{x=1}^{C}\frac{\text{TP}(x)}{\text{TP}(x) + \text{FP}(x) + \text{FN}(x)}. \quad (1)$$

4.1. Cityscapes. Cityscapes is a large dataset with video sequences recorded from the streets of 50 different cities. The dataset has 2975 training samples, 500 validation samples, and 1525 test images. We trained our networks at an image resolution of 256×256 and with 10 classes. To test the performance of both multipath feature extraction methods, we have trained our network first with a PSP module as our decoder and then with an ASPP module. Table 3 shows the comparison between ENet, SINet, and our proposed networks on the CityScapes dataset, the number of parameters of SINet are still much less than our proposed architecture but the jump in mIoU is significant while still using half trainable parameters; the graphical comparison between the models is shown in Figure 5 shows that both of our models tend to converge a bit sooner. Despite having slightly more parameters than the PSP module, the ASPP module is still faster than the PSP module while still producing better results.

The results in Figure 6 show that high-resolution feature encoding in our model makes it better at segmenting thin and small objects and also at predicting fine-edged feature masks. The pole in the first example is segmented by both of our networks with reasonable accuracy while ENet and SINet completely ignored it. The person in the second image is segmented as a blob by SINet and ENet, whereas our proposed architectures have managed to produce better edges and a more human-like shape.

4.2. Mapillary Vistas. Mapillary Vistas has 25,000 high-resolution images which are 5 times larger than Cityscapes; it contains 66 object categories with labels for 37 classes. It contains images from all devices from all around the world in various weather conditions and seasons. We have augmented our dataset by flipping the images along the y-axis, doubling the dataset. We divided our dataset so that we had 40,000 training samples, 5,000 validation, and 5,000 test samples. Table 4 shows how each network performed on the Mapillary dataset. Figure 7 shows that our architectures maintain a pattern similar to the one presented by Figure 5 on a much larger Mapillary vistas dataset.

All 4 networks are trained on 3 classes naming vehicle, pedestrian, and road. Figure 8 shows the visual comparison between the results of all 4 networks on Mapillary Vistas.

Both of our networks have shown similar improvements on both datasets. Careful encoding of features in high resolution combined with multipath feature extraction has shown to segment finer edges without any increase in the number of learnable parameters. The first example in Figure 8 shows how both of our M-Net architectures were able to segment 3 different cars separately instead of segmenting all three cars as one. In the second example both M-Nets are able to produce much finer results showing how it is able to segment both large and small objects. The graphs below from (a) to (d) show the change in mIoU with every epoch on a validation set.

5. Discussion

This paper improves on the traditional encoder-decoder technique for segmentation and proposes a technique to encode the features in full resolution and uses a multipath

feature extraction feature extraction module to predict much finer segmentation masks as compared to its traditional encoder-decoder counterparts.

U-Net [6] has been one of the most widely used encoder-decoder architecture for semantic segmentation; its effectiveness and simplicity is the main reason behind its popularity. It is safe to say that aggressive down sampling in segmentation models can cause the loss of important spacial information. It can be argued that the skip connections in U-Net [6] and SegNet [13] can overcome the loss of information due to downsampling, but looking at it from a different angle, it is clear that the convolutional layer immediately after the pooling layer will not receive the needed spacial information. Small models like ENet [24] and SINet [25] downsample the features in the beginning of the network and then go deep with much smaller feature maps to reduce the size and computational requirements of the model. In this paper, we show why that is not a good idea when network is to be used for road scene segmentation. Encoding the features in full resolution and using a multipath feature extraction module has shown to result in much finer and accurate segmentation masks while still maintaining low computational requirements. The future work of this study may include upscaling the network to compare its performance with larger segmentation models. The main limitation of this technique is that going too deep with full scale features can be expensive this is one of the reasons why we had to use max-pooling to be better than the networks under consideration (ENet and SINet) in both size and speed. Future work might also be able to study the effect of going deeper with full scale features for applications where computation resource is not an issue.

6. Conclusion

Unlike detection and classification applications, spacial resolution of features is extremely important when it comes to segmentation. This is also true for road scenes when segmenting small objects like a person and traffic sign, etc. This paper proposes a new deep learning-based model for semantic segmentation using an encoder-decoder architecture.

Instead of following the conventional approach of doing extensive downsampling of features in the encoder, we have introduced the idea of high-resolution feature encoding, thus enabling the decoder to extract valuable multiscale features from the high-resolution encoded features. To address the issue of latency due to high-resolution features, the spatial resolution is reduced by half after every two convolution blocks. The downsampled features are then upsampled before being concatenated with the rest of the features. This way the output of the encoder is in full resolution. The decoder consists of a multipath feature extraction module to decode the necessary information from three different scales. The proposed scheme is also compared with some classical encoder-decoder architectures for semantic segmentation. The experimental results reported in the paper show that encoding in full resolution has resulted in the prediction of much finer segmentation masks for both large and small objects. This research shows the overall effectiveness of the proposed architecture in terms of improved segmentation performance.

Data Availability

Two datasets were used in this set of experimentation namingly Cityscapes and Mapillary vistas; they both are open access datasets and are available on the following links: https://www.kaggle.com/datasets/zhangyunsheng/cityscapes-data and https://www.mapillary.com/dataset/vistas.

Conflicts of Interest

There are no potential conflicts of interest. The work has been undertaken to accept standards of ethics and of professional standards.

Acknowledgments

One of the authors (Harish Kumar) extends his gratitude to the Deanship of Scientific Research at King Khalid University for funding this work through research groups program under grant number R. G. P. 2/198/43.

References

[1] M. Bojarski, D. Del Testa, D. Dworakowski et al., "End to end learning for self-driving cars," 2016, http://arxiv.org/abs/1604.07316.

[2] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, http://arxiv.org/abs/1804.02767.

[3] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2020.

[4] Z. Li, G.-A. Bilodeau, and W. Bouachir, "Multiple convolutional features in Siamese networks for object tracking," *Machine Vision and Applications*, vol. 32, no. 3, pp. 1–11, 2021.

[5] T. Darrell, J. Long, and E. Shelhamer, "Fully fonvolutional networks for semantic segmentation," 2015, http://arxiv.org/abs/1411.4038.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," 2015, http://arxiv.org/abs/1505.04597.

[7] 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, 2018.

[8] R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka, "Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1442–1450, 2018.

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

[10] Z. Shao, Z. Zhou, X. Huang, and Y. Zhang, "MRENet: simultaneous extraction of road surface and road centerline in complex urban scenes from very high-resolution images," *Remote Sensing*, vol. 13, no. 2, p. 239, 2021.

[11] T. Emara, H. E. Abd, E. Munim, and H. M. Abbas, "LiteSeg: a novel lightweight ConvNet for semantic segmentation," in

2019 Digital Image Computing: Techniques and Applications (DICTA), 2019.

[12] V. B. A. Kendall and R. Cipolla, "Bayesian SegNet: model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *Proceedings of the British Machine Vision Conference (BMVC)*BMVA press.

[13] 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, no. 12, pp. 2481–2495, 2017.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image," 2015, http://arxiv.org/abs/1409.1556.

[15] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "FastFCN: rethinking dilated convolution in the backbone for semantic segmentation," 2019, http://arxiv.org/abs/1903.11816.

[16] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: multi-path refinement networks for high-resolution semantic segmentation," 2016, http://arxiv.org/abs/1611.06612.

[17] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," 2018, http://arxiv.org/abs/1804.09337.

[18] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, http://arxiv.org/abs/1706.05587.

[19] P. Shamsolmoali, M. Zareapoor, H. Zhou, R. Wang, and J. Yang, "Road segmentation for remote sensing images using adversarial spatial pyramid networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 4673–4688, 2021.

[20] S. Xie, H. Hu, and Y. Wu, "Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition," *Pattern Recognition*, vol. 92, pp. 177–191, 2019.

[21] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," 2020, http://arxiv.org/abs/2005.10821.

[22] M. Yap, *Road damage segmentation for mobile hardware, [M.S. thesis]*, KTH Royal Institute of Technology, Stockholm, Sweden, 2021.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[24] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: a deep neural network architecture for real-time semantic segmentation," 2016, http://arxiv.org/abs/1606.02147.

[25] H. Park, L. L. Sjösund, Y. Yoo, N. Monet, J. Bang, and N. Kwak, "SINet: extreme lightweight portrait segmentation networks with spatial squeeze modules and Information Blocking Decoder," 2020, http://arxiv.org/abs/1911.09099.

[26] Y. Li, X. Zhang, and D. Chen, "CSRNet: dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1091–1100, 2018.

[27] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2017, http://arxiv.org/abs/1412.6980.

[28] M. Cordts, M. Omran, S. Ramos et al., "The Cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[29] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The Mapillary Vistas dataset for semantic understanding of street scenes," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5000–5009, 2017.