IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 7, 2021, accepted January 15, 2021, date of publication January 19, 2021, date of current version January 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052791

REF-Net: Robust, Efficient, and Fast Network for Semantic Segmentation Applications Using Devices With Limited Computational Resources

BEKHZOD OLIMOV, JEONGHONG KIM, AND ANAND PAUL, (Senior Member, IEEE)
School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Jeonghong Kim (jhk@knu.ac.kr)

This work was supported in part by study was supported by the School of Computer Science and Engineering, Ministry of Education, Kyungpook National University, South Korea through the BK21 Four project (AI-driven Convergence Software Education Research Program) under Grant 4199990214394, and in part by the National Research Foundation of Korea under Grant 2020R1A2C1012196.

ABSTRACT Considering importance of the autonomous driving applications for mobile devices, it is imperative to develop both fast and accurate semantic segmentation models. Thanks to emergence of Deep Learning (DL) techniques, the segmentation models enhanced their accuracy. However, this improved performance of currently popular DL models for self-driving car applications come at the cost of time and computational efficiency. Moreover, networks with efficient model architecture experience lack of accuracy. Therefore, in this study, we propose robust, efficient, and fast network (REF-Net) that combines carefully formulated encoding and decoding paths. Specifically, the contraction path uses mixture of dilated and asymmetric convolution layers with skip connections and bottleneck layers, while the decoding path benefits from nearest neighbor interpolation method that demands no trainable parameters to restore original image size. This model architecture considerably reduces the number of trainable parameters, required memory space, training, and inference time. In fact, the proposed model required nearly 90 times fewer trainable parameters and approximately 4 times less memory space that allowed 3-fold faster training runtime and 2-fold inference speedup in the conducted experiments using Cambridge-driving Labeled Video Database (CamVid) and Cityscapes datasets. Moreover, despite its notable efficiency in terms of memory and time, the REF-Net attained superior results in several segmentation evaluation metrics that showed roughly 2%, 4%, and 3% increase in pixel accuracy, Dice coefficient, and Jaccard Index, respectively.

INDEX TERMS Autonomous driving, deep convolutional neural networks, nearest neighbor interpolation, semantic segmentation.

I. INTRODUCTION

Being one of the most popular members of computer vision tasks, semantic segmentation has been widely used in numerous applications in various domains. Although traditional methods for semantic segmentation mainly depended on domain expert intervention, heavy use of high level engineering skills for feature choice [1], emergence of DL techniques entailed unprecedentedly notable progress in a number of semantic segmentation fields, namely, medicine [2]–[4], biomedicine [5]–[7], geo-sensing [8]–[10], fashion [11]–[13], and autonomous driving [14]–[16].

An autonomous driving (driverless) vehicle is a means of transport that possesses ability to recognize its surroundings and move safely with little or no human intervention [17], [18]. Depending on the human intervention in driving process, autonomous vehicles are categorized into five levels. As the level progresses, the human intervention becomes less involved. Specifically, Level 0 vehicles are under full control of a driver and Level 5 vehicles are totally independent from human activity [19].

The associate editor coordinating the review of this manuscript and approving it for publication was Qi Zhou.

A successfully launched project using 5-ton VaMoRs van in the 90s of XX century as well as rural and urban self-driving car challenges organized by Defense Advanced Research Projects Agency (DAPRA) in 2005 and 2007 gave great impetus to development of autonomous driving research that attracted researchers all over the world to contribute to the improvement of this field [20]. The other reasons for high interest in the area of autonomous driving are related to the environment protection and customer satisfaction. Specifically, self-driving cars possess great positive influence on addressing the problems of carbon emissions as well as traffic jam and driver safety.

Although autonomous driving entails numerous optimistic consequences, it is a highly difficult task to develop robust self-driving vehicle system due to complicated and unexpected situations in urban areas. These factors made environment perception of autonomous vehicles challenging. In fact, autonomous vehicle percepts and recognizes its surroundings using various sensors, namely, radar, lidar, and camera. Therefore, improving the perception ability of the vehicles was the broad and active research area so far. Particularly, great number of research works focused on radar-based [21]–[23], lidar-based [24]–[28], and camera-based [29]–[32] object detection.

Based on self-driving car characteristics, a system to perceive its surroundings is expected to be real-time, which requires time-efficient models. Also, considering autonomous vehicle involves little or no human intervention safety of its passengers is crucial. Moreover, autonomous driving cars should be robust to adverse weather conditions that can make specific sensors of the vehicles defective or out of order. Consequently, a state-of-the-art model for autonomous driving is required to be fast, accurate, and robust. However, study of the existing methods showed that to obtain high accuracy they require enormous number of trainable parameters [5], [33]–[36], which result in slow training and inference speed. These factors make self-driving cars environment perception challenging and lead to unsafe driving. Moreover, owing to large size of the models, it is difficult to use them for mobile or battery-powered applications [37]. Therefore, in this study, we propose REF-Net that requires considerably fewer parameters; consequently, needs less time for training and inference. In addition, the model can obtain significantly better and more accurate performance in comparison with the existing efficient methods.

REF-Net benefits from threefold residual networks in the contraction and a new upsampling technique in the expansion paths. Thanks to usage of bottleneck layers in the contraction and nearest neighbor interpolation upsampling method in the expansion paths, the model requires significantly fewer parameters to train the model, which results in speed-up in training and inference phases.

In fact, the proposed model addresses the aforementioned existing problems and contributes to enhancing the autonomous driving field in the following ways:

- The REF-Net model introduces residual skip connections along with bottleneck layers in the contraction part and nearest neighbor interpolation upsampling method used in the expansion path, which allow the proposed model to outperform the existing expensive networks in terms of computation and time.
- Although the REF-Net demands fewer trainable parameters, the proposed method obtains the-state-of-the-art performance when assessed with several evaluation metrics, namely, pixel accuracy, mean IoU and Dice coefficient. Therefore, this study provides beneficial guidelines to fine-tune parameters and model architecture to obtain accurate semantic segmentation results.
- The REF-Net model produces the segmented autonomous driving images 2 × faster when compared with the exiting computationally expensive methods. To the best of our knowledge, there has been no such a fast and accurate proposed yet. Thus, the REF-Net can be used as the benchmark in the related autonomous driving semantic segmentation research.
- The REF-Net addresses the issue of introducing semantic segmentation tasks in mobile or battery-powered applications due to its speed and accuracy. Also, real-time autonomous driving applications may hugely benefit from the proposed model owing to small size, fast performance and accurate segmented results.

The manuscript is organized in as follows. In Section II, we present an overview of the existing methods related to the semantic segmentation in autonomous driving field. Section III provides detailed information on the proposed REF-Net method. Section IV provides the details of the conducted experiments and results using the considered models. Section V discusses the experimental results. Finally, Section VI concludes the research and outlines potential future research directions.

II. RELATED WORK

In this section, we describe currently available methods for semantic segmentation in driverless vehicles domain. For convenience, we divide the techniques into two broad groups, such as traditional methods and DL methods.

A. TRADITIONAL METHODS FOR SEMANTIC SEGMENTATION IN AUTONOMOUS DRIVING FIELD

Before the emergence of DL, there were three most widely used methods for semantic segmentation, namely random forest classifier (RFC), conditional random fields (CRF), and boosting. Although these approaches were not able to attain desired accuracy results, they could obtain the state-of-the-art performance approximately a decade ago. Specifically, Shotton *et al.* proposed efficient and powerful low-level features, called semantic texton forests that used ensembles of decision trees and act directly on image pixels [38]. The texton forests did not require computation of filter-bank responses; thus, were very fast in both train and test stages. Also, Brostow *et al.* developed a semantic segmentation

algorithm using 3D point clouds obtained from ego-motion [39]. The authors projected 3D cues on 2D image plane by modeling spatial layout and context simultaneously. After obtaining the features, randomized decision forest was exploited to generate a precise 2D segmentation and classify objects into pre-defined categories.

One of the most prominent approaches for traditional sematic segmentation was developed by Sturgess *et al.* [40]. The proposed method combined appearance and structure from motion features. Label likelihoods were modeled using CRF framework and the a priori knowledge. Also, textons, color, location-based features were used as an input to a novel boosting algorithm that obtained the-state-of-the-art performance on CamVid. Further, Ladicky *et al.* proposed a hierarchical random field model that combined various features obtained from different stages of quantization hierarchy [41]. Due to usage of powerful graph cut-based algorithms, the proposed model was very efficient in inference and showed better generalizability than the existing approaches. Similarly, Kohli *et al.* [42] introduced a method by addressing the labeling problem using higher order CRF. This technique allowed to partially alleviate misleading segments that spanned multiple object categories. Also, the authors exploited higher order potentials, which assisted to group the pixels within a single segment to share the same label.

Regarding boosting approaches, Shotton *et al.* introduced TextonBoost algorithm learned a discriminative model of object classes based on shape, appearance, and context information [43]. The authors benefited from the discriminative model that used features based on textons. Owing to usage of random feature selection and piecewise training methods, this method trained a model in an efficient way and obtained competitive results in segmenting highly textured, highly structured, and articulated objects. He *et al.* [44] developed a technique that comprised contextual features for labeling images, where each pixel belonged to one of the categories. These features were combined into a probabilistic framework that incorporated the outputs of various components that focused on the image-label mapping and patterns within the label.

B. DL METHODS FOR SEMANTIC SEGMENTATION IN AUTONOMOUS DRIVING FIELD

Despite being used for long time, traditional approaches required heavy hand engineering as well as were time consuming and not accurate. Therefore, after introduction of DL approaches that were efficient in terms of time and obtained notable performance, traditional methods became nearly obsolete. Consequently, being a member of DL techniques, Deep Convolutional Neural Networks (DCNN) are heavily utilized in semantic segmentation for the last few years. DCNN models architecture comprises encoding (contraction) and decoding (expansion) paths. In the encoding part, the features of an image are extracted by decreasing an image size is decreased and increasing of its depth. However, at this stage a model has insufficient information about the location of the features. Therefore, in the decoding part, an image is restored to its original size by decreasing an image depth, which allows a model to learn the location of the features.

In the early stages of semantic segmentation using DL methods, the researchers mainly segmented LiDAR points were utilized to segment LiDAR points [45]–[47].

One of the most popular approaches for semantic segmentation is single-stage pipeline-based fully convolutional network. Long *et al.* proposed a DL model architecture containing only convolutional operations [48]. Badrinarayanan *et al.* proposed another DCNN model called SegNet that contained encoder and corresponding decoder networks followed by a pixel-wise classification layer [33]. The authors used VGG16 [49] network as an encoder model and in the decoding part the encoded image was transformed into the original one that restored the low resolution features maps as full resolution feature maps. This was obtained by usage of pooling indices saved from the maxpooling operation during the encoding path, which eliminated training in the expansion part. The resulted upsampled sparse feature maps then convolved with trainable filters to generate dense feature maps. The model attained notable accuracy results in comparison to the existing models at that time.

One of the most notable DCNN models for semantic segmentation is U-Net [5]. The model was originally proposed for binary segmentation in medicine domain; however, owing to its generalizability, it was utilized in multiclass semantic segmentation in various fields. Its architecture was symmetrical and contained four large blocks in the encoding and four large blocks in decoding as well as one bridge block that connects encoding and decoding parts. The contraction part of the model executed two convolution operations with the filter size of 3×3 followed by a 2×2 max pooling layer and dropout regularization. Concerning the expansion path, after increasing the image size through transpose convolution, the output was concatenated using the corresponding output of the extraction path convolution layer. Then the result passed through the same set of operations as in the expansion path apart from the max pooling operation. Finally, when the size of the original image was restored, one filter using a kernel with the size of 1×1 executed the convolution operation to produce a resulting segmented image.

Jegou *et al.* presented fully convolutional dense network for semantic segmentation [34] based on densely connected convolutional networks [50] that ensured that each layer of the model was directly linked to the other layers of the network in encoding path. Concerning the restoring an original input image, transpose convolution was used in the expansion path. In fact, the model architecture contained 103 convolutional layers. Despite of comprising great number of layers, the network obtained the-state-of-the-art performance in multiclass semantic segmentation owing to the usage of the densely connected convolutional neural network in the contraction path. Also, the fine-tuned model eliminated need for post-processing module to achieve better performance.

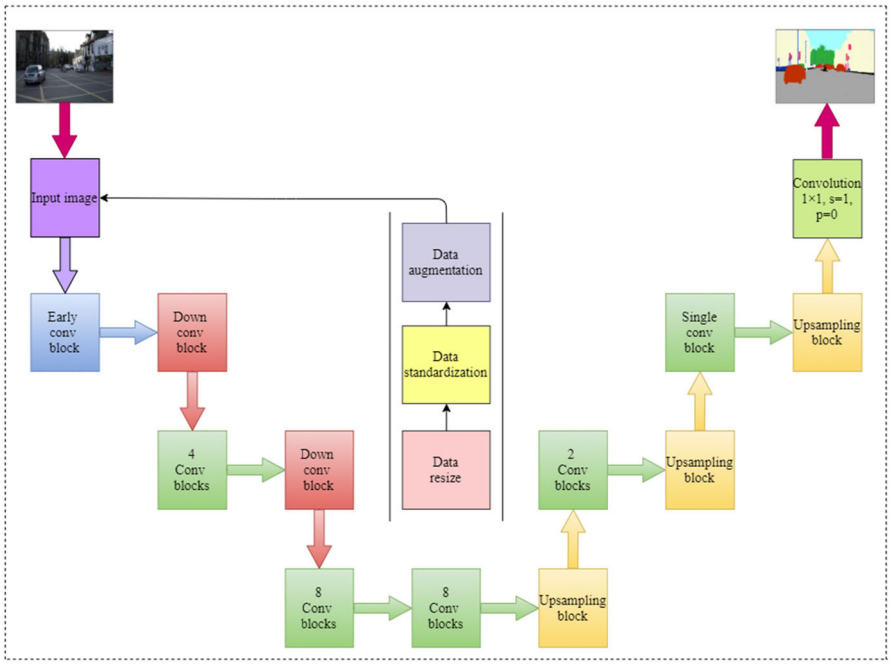

FIGURE 1. An overview of the proposed methodology.

However, the aforementioned models required large number of parameters for training that resulted in slow training and inference by making these models challenging to use for computationally limited device applications. Considering this fact, Pazske *et al.* presented efficient DCNN for real-time semantic segmentation [37]. The authors benefitted from using various convolution operations in the decoding path, such as asymmetric and dilated convolution. Also, they utilized bottleneck layers inspired from [51]. Regarding decoding part of the model, it used convolution transpose operation to restore an original image. The model was enormously efficient in comparison with the existing methods by requiring significantly few trainable parameters.

III. THE PROPOSED METHODOLOGY

This section contains comprehensive information about the REF-Net model and its architecture. An overview of the proposed model is represented in Figure 1. As can be seen from the graphical illustration of the proposed model, first, the input data for training passes through three-step data preprocessing. At the initial stage, the data is resized to match an input size of the model. Then, data standardization is applied to make the values of the input image follow Gaussian normal distribution. This is obtained by subtracting mean value of the data (μ) and dividing into the standard deviation value (σ) as shown in Equation (1):

$$X = \frac{X - \mu_X}{\sigma_X} \tag{1}$$

After obtaining the standardized images, data augmentation is applied on them to increase the number of training instances. The data augmentation is performed based on the characteristics of the images, meaning that there is no rule of thumb that work properly for all data.

Completing the preprocessing steps, the data is ready to be inputted into a model. It comprises two parts, namely, encoding (contraction) and decoding (expansion) paths. In the encoding part, the model learns useful features, parts of objects, and complete objects by decreasing the image size and increasing the image depth as the training process progress. On the contrary, the model attempts to identify location of these objects by restoring the original image size in the decoding part. It is important to note the computation of an image size in a convolution layer. It can be calculated using Equation (2) as follows:

$$\begin{aligned} H_I &= \frac{H_I - fs + 2*p}{s} + 1 \\ W_I &= \frac{W_I - fs + 2*p}{s} + 1 \end{aligned} \tag{2}$$

In Equation (2), H_I, W_I are the height and width of an image, f_s is the kernel size of a convolution operation filter, p is zero-padding, and s is stride that is responsible for a step of the convolution filter.

The contraction path contains an early conv block that has two branches. The first one performs convolution operation using 13 3 $\times$ 3 filters with stride of 2 and zero-padding to decrease the image size by two times. The second branch performs overlapping maxpooling operation with stride of 2 to match the image size from the first branch of the early conv block. Then, the outputs of these branches are concatenated

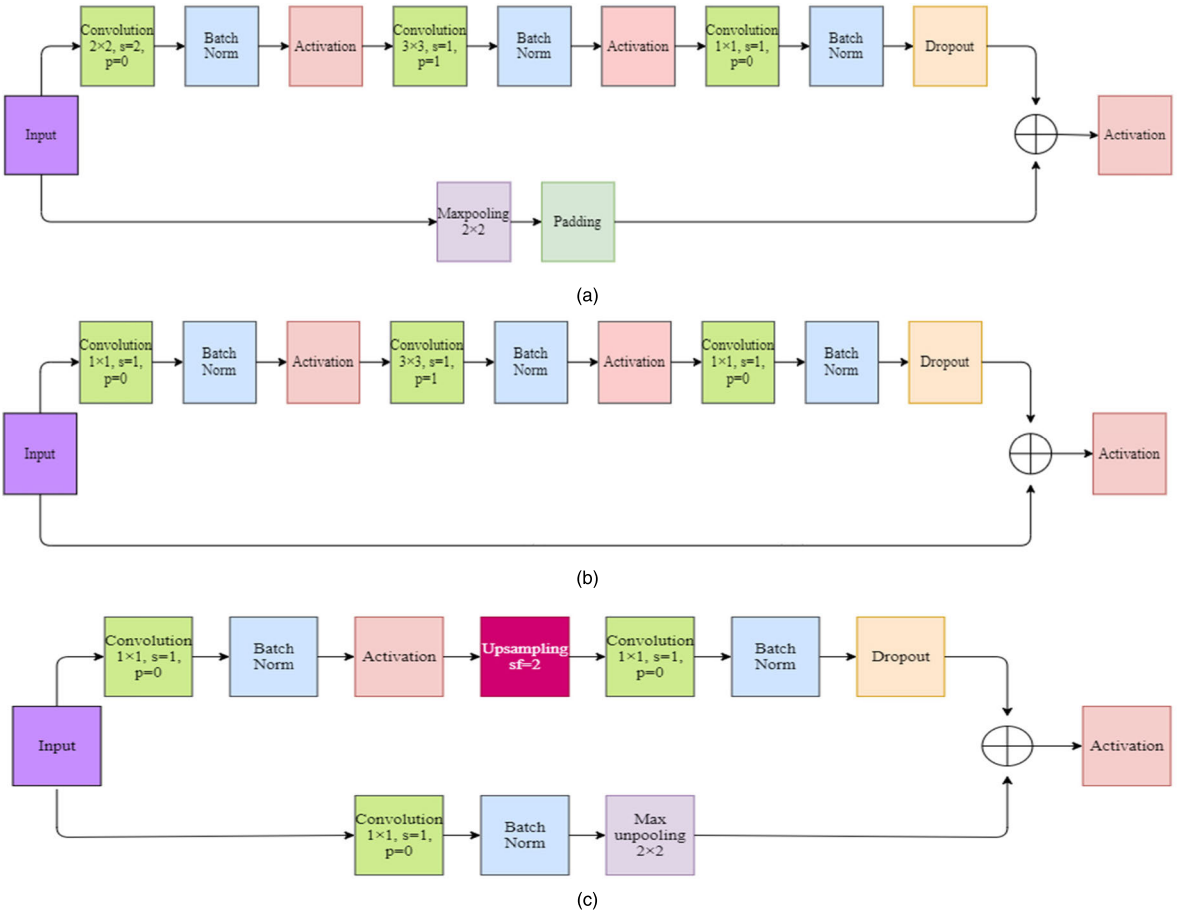

FIGURE 2. Detailed description of (a) the down conv blocks, (b) the conv blocks, and (c) upsampling blocks.

followed by batch normalization [52] and parametric rectified linear unit (PReLU) [53] activation layers.

The output of the early conv block then passes through downsampling conv block illustrated in Figure 2 (a). Observably, the down conv block is highly inspired by skip connections [54] that have wide range of applications in various domains. However, the difference of the proposed method with the regular residual networks is that we utilized 2×2 filters instead of using default 1×1 filters in the first convolution layer that provided better performance in [37]. Also, the first branch of the block contains dropout regularization [55] to address overfitting after repetitive convolution, batch normalization, and activation layers. Moreover, instead of a regular convolution layer with stride of 2, the second branch contains maxpooling layer followed by padding to match the image size of the first branch output.

This step is applied to reduce the number of trainable parameters; consequently, reduce computational complexity and training time. Finally, the outputs of the two branches are added and pass through an activation layer.

After obtaining the output of the down conv block, the data goes through several conv blocks, which are illustrated in Figure 2 (b). The conv blocks have similar structure to the down conv blocks with slight differences in the first convolution layer of the first branch and complete second branch. Specifically, we used 1×1 filters with stride of 1 as the first convolution operation of the conv block instead of 2×2 filters with stride of 2 used in the down conv block. Since the image size is not changed in the first branch of the conv block, we do not apply any operation in the second branch. The conv blocks are completed by adding the outputs of the branches and going through an activation layer. It is notable that for the 5∼20th conv blocks, main convolution layer of them benefit from the mixture of asymmetric convolution layers [56] as the main convolution layer followed by dilated convolution layers [57], where dilation rate increases every following layer.

Combination of the early conv block, 2 down conv blocks, and 20 conv blocks finishes the encoding part of the model and its output is inputted into the decoding path. In the expansion path, we use nearest neighbors upsampling method in contrast to the transposed convolution utilized in the existing methods. We found transposed convolution approach expensive in terms of both computation and time because it requires additional parameters that are trained using an optimizer that demands computation of gradients of these parameters

in every step of backpropagation algorithm. As the primary objective of this study is to develop fast and accurate semantic segmentation model architecture, we utilized inexpensive method for restoring the image size that is nearest neighbor interpolation upsampling method. The computation method of the aforementioned techniques for upsampling are provided in Equation (3), as shown at the bottom of the next page.

In the equation, x is a pixel of an input image for upsampling, w is a weight parameter used to increase the size of an image using transposed convolution, y is a pixel of an upsampled image. As can be seen, the transposed convolution requires additional trainable parameters to perform upsampling, which are optimized as training stage progresses. Moreover, input data values are multiplied by weight parameters that lead to computational complexity and increase in training time. In contrast, the nearest neighbor interpolation method simply copies the values of the input image into the output matrix. Although this approach is simple, it is completely logical since an image contains hundreds of pixels and the neighboring pixels are approximately the same in most cases. More importantly, this technique requires no additional parameters nor extra computation, which are crucial factors in dealing with the existing problems in real-time semantic segmentation. Also, this method obtains competitive accuracy, which will be discussed in the results section of this manuscript.

Selecting the nearest neighbor interpolation method for recovering the image size in the decoder part of the model, we provide detailed graphical illustration of the upsampling block in Figure 2 (c).

Observably, upsampling blocks use the power of skip connections too. However, unlike to the existing methods, upsampling method is not performed right after obtaining the input data from the previous layers in the upper branch. In the proposed method, we first perform convolution operation with 1×1 filters and then increase the image size by factor of 2 using the nearest neighbor interpolation method followed by 1×1 convolution, batch normalization, and dropout. This strategy assists to reduce computational complexity by applying 1×1 convolution to reduce the depth of an incoming image.

The lower branch also benefits from this trick by using 1×1 convolution followed by batch norm and maxunpooling operation that uses the indices from the maxpooling operation of the encoding path to increase image dimensions. After increasing the image size by factor of 2 in both branches, they are added and pass through an activation layer to produce an output. This output then goes through two conv blocks, upsampling block, conv block, respectively. Finally, a segmented image is generated using convolution operation with 1×1 filter.

IV. EXPERIMENTS AND RESULTS

In this section, we provide comprehensive information about conducted experiments to test the performance of the proposed model on two publicly available autonomous driving datasets. Also, we share outcomes of these experiments and compare them with the results of the existing methods.

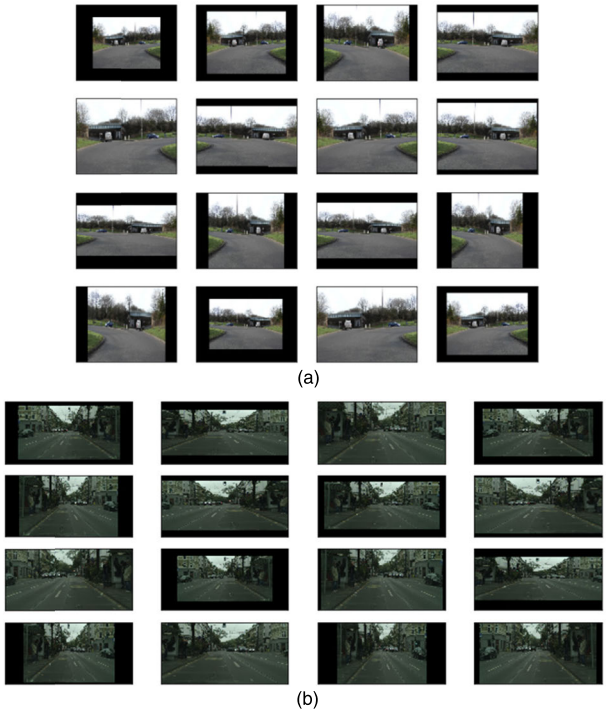

FIGURE 3. Randomly selected training image from (a) CamVid and (b) Cityscapes datasets after applying data augmentation.

TABLE 1. Detailed description of the datasets used for the experiments.

Dataset Names	CamVid	Cityscapes
Image Type	Urban Street Scenes	Urban Street Scenes
Image Size	360 × 480	360 × 480
Categories	12 *	30 **
Train Images	367	2975
Validation Images	101	500
Test Images	233	1525

* Smaller version of the dataset with 12 classes is used for the experiments.
** 19 out of 30 classes are used for semantic segmentation as in [58].

A. DATASETS

We used two popular databases that are in open access for research and widely used to evaluate model performance in self-driving cars field. The first database was CamVid and the second one was Cityscapes dataset. Table 1 represents general overview of them.

We can obtain from Table 1 that both datasets comprised considerable large sized images. Therefore, to reduce computational expenses, we resized the input images into 360×480 and 512×1024 for CamVid and Cityscapes datasets, respectively. Also, the datasets contained limited number of examples for training and validation sets. We addressed this issue by generating new training and validation images by applying data augmentation. Specifically, we applied horizontal flip and zoom scaling in the range of 0.75∼1.5 to obtain transformed images. The output of this process on

random training images from the considered datasets are provided in Figure 3.

B. BASELINE MODELS

We selected four well-known semantic segmentation models, namely SegNet, FC-DenseNet103, U-Net, and ENet, to compare their performance with the one of the proposed method. Since these models were described in detail in the introduction part of this manuscript, we do not go deep into their specifications in this section.

The first three models were chosen as the networks that obtain state-of-the-art performance, but expensive in terms of time and computation. Meanwhile, ENet was selected as a time-efficient model that achieves comparatively limited accuracy. By selecting these models, we wanted to compare the performance of the proposed method with regard to both computation time and accuracy.

C. TRAINING SETUP

We formulated the proposed method as well as the baseline models using 3.6.9 version of Python as well as 1.4.0 version of PyTorch framework and conducted experiments using 32 GB NVIDIA Tesla V100-SXM2 GPU with CUDA 10.0. In all experiments, we initialized the weight parameters using Kaiming weight initialization strategy [53] and Adam optimizer [59] with a momentum of 0.9, learning rate of 5e-4, weight decay of 2e-4. As for the loss function to minimize, we used crossentropy loss. We trained the models for a hundred epochs with a batch size of 10 and 4 for CamVid database and Cityscapes dataset, respectively. This number of epochs was selected because in average the models converged at epoch 100 and stopped improving their performance afterwards.

D. EVALUATION METRICS

Most datasets used for semantic segmentation exhibit a problem of class imbalance, where one category has significantly higher rate of representation in comparison with the other classes. Therefore, being most widely used evaluation metric, accuracy, cannot fairly assess the performance of a segmentation model. Considering this fact, we conducted evaluation of the models using not only pixel accuracy (PA) metric but also Dice coefficient (DC) and mean intersection over union (mean IOU). The PA computes average score of the ratio of correctly predicted pixels with regard to target pixels as follows:

$$PA = \frac{1}{m}\sum_{i}^{m}\frac{\sum_{k}^{p}\hat{y}_k == y_k}{\sum_{k}^{p} y_k} \tag{4}$$

In Equation (4), $\hat{y}$ and y are predicted and target values, p and m are total number of pixels in an image and total number of instances, respectively.

Dice coefficient computes twice of intersection area of two images divided by the area of their union and formulated in a following way:

$$DC_{A,B} = 2\frac{|A \cap B|}{|A| + |B|} \tag{5}$$

Regarding mean IoU, it calculates the ratio of overlapping area between two images with the intersected area subtracted from their union as shown in Equation IV-E:

$$mean\ IoU_{A,B} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{6}$$

E. EXPERIMENTAL RESULTS

In this sub-section, we share the results of the experiments using the CamVid database and Cityscapes dataset. First, we compare the models memory requirements and the number training parameters in Table 2.

As it can be seen from Table 2, the proposed model required the fewest number of trainable parameters when compared with the baseline models by demanding nearly 90, 85, and 27 times fewer parameters than U-Net, SegNet, and FC-DenseNet103 models, respectively. Moreover, the proposed model needed more than 100 times less memory to store trainable parameters in contrast to SegNet and U-Net models. The only model that could compete with REF-Net is ENet, which was also less efficient in terms of memory and trainable parameters in comparison to the proposed model.

In fact, considering the baseline models, the REF-Net model demanded the least memory space and the fewest number of trainable parameters.

We also compared the considered models in terms of time required for training using CamVid and Cityscapes datasets.

$$\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \xrightarrow{\substack{transposed \\ convolution}} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \\ y_5 & y_6 & y_7 & y_8 \\ y_9 & y_{10} & y_{11} & y_{12} \\ y_{13} & y_{14} & y_{15} & y_{16} \end{bmatrix} \xrightarrow{computed\ as} \boxed{\begin{array}{l} y_1 = x_1 * w_1 \\ y_2 = x_1 * w_2 \\ \cdots \\ y_{15} = x_4 * w_3 \\ y_{16} = x_4 * w_4 \end{array}}$$

$$\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \xrightarrow{\substack{nearest\ neighbor \\ interpolation\ upsampling}} \begin{bmatrix} x_1 & x_1 & x_2 & x_2 \\ x_1 & x_1 & x_2 & x_2 \\ x_3 & x_3 & x_4 & x_4 \\ x_3 & x_3 & x_4 & x_4 \end{bmatrix} \tag{3}$$

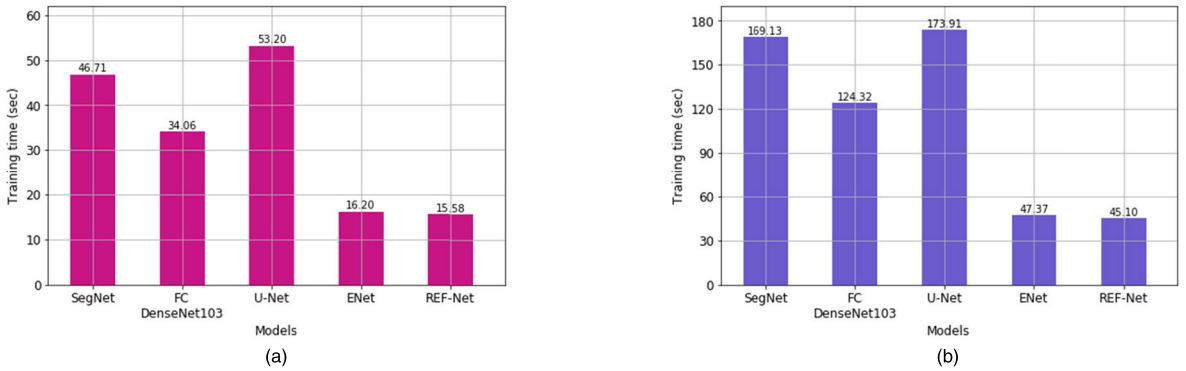

FIGURE 4. Required average time per epoch to train the considered models using (a) CamVid and (b) Cityscapes datasets.

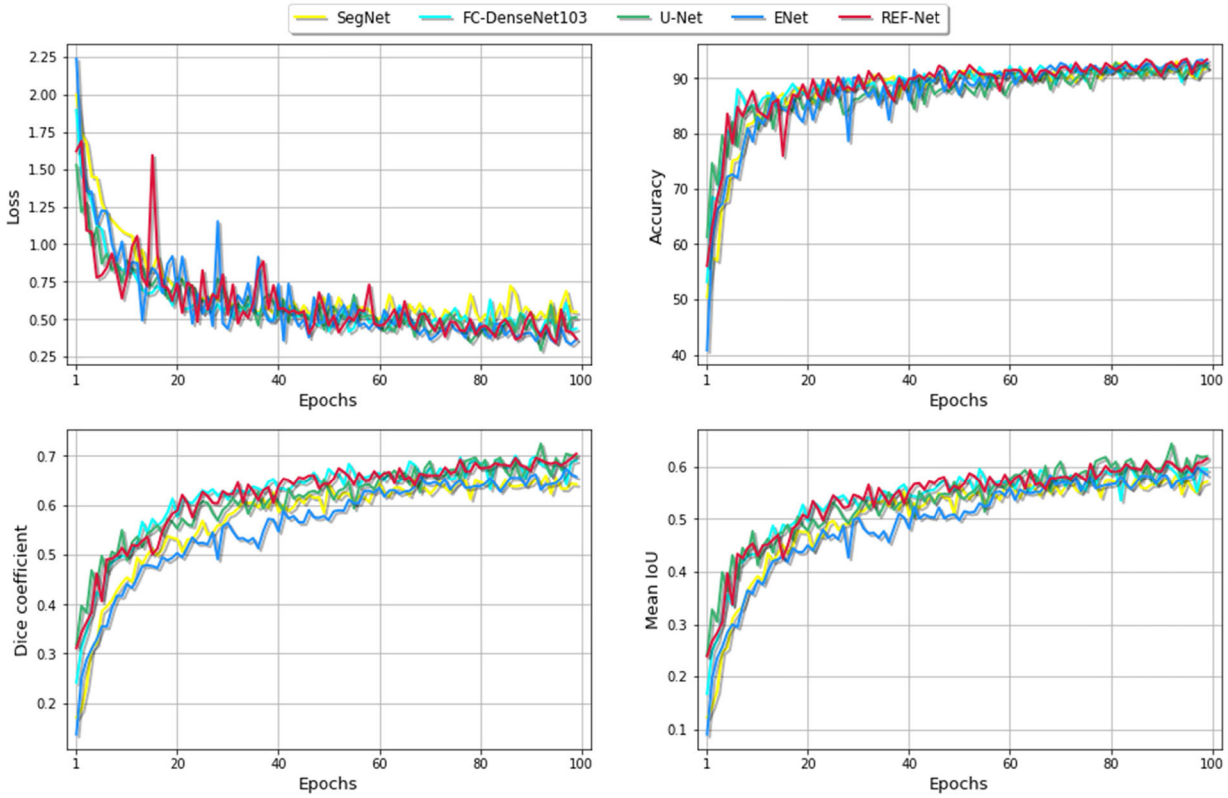

FIGURE 5. Experimental results on the validation set of CamVid dataset.

Bar chart illustrated in Figure 4 shows average time per epoch required to train the models on the aforementioned datasets. Logically, the models with the least number of trainable parameters, such as ENet and REF-Net significantly outperformed their peers, namely SegNet, FC-DenseNet103, and U-Net, that required from 3 to 4 times greater amount of time to be trained.

After obtaining the results of the aforementioned models with reference to memory, parameters, and training time, we compared them in terms of accuracy using the three evaluation metrics mentioned above and loss value. Figure 5 illustrates loss, PA, DC, and mean IoU values on the CamVid's validation set. We can see that the SegNet and ENet models obtained relatively lower results than the other models. In fact, the performances of U-Net and REF-Net were superior to their peers in terms of considered evaluation metrics on the validation data. At the same time, the FC-DenseNet103 model attained slightly higher loss and negligibly lower scores in the accuracy metrics in comparison with these two models. Considering the REF-Net model's efficiency in computation, memory, and time, the proposed model obtained relatively competitive accuracy compared

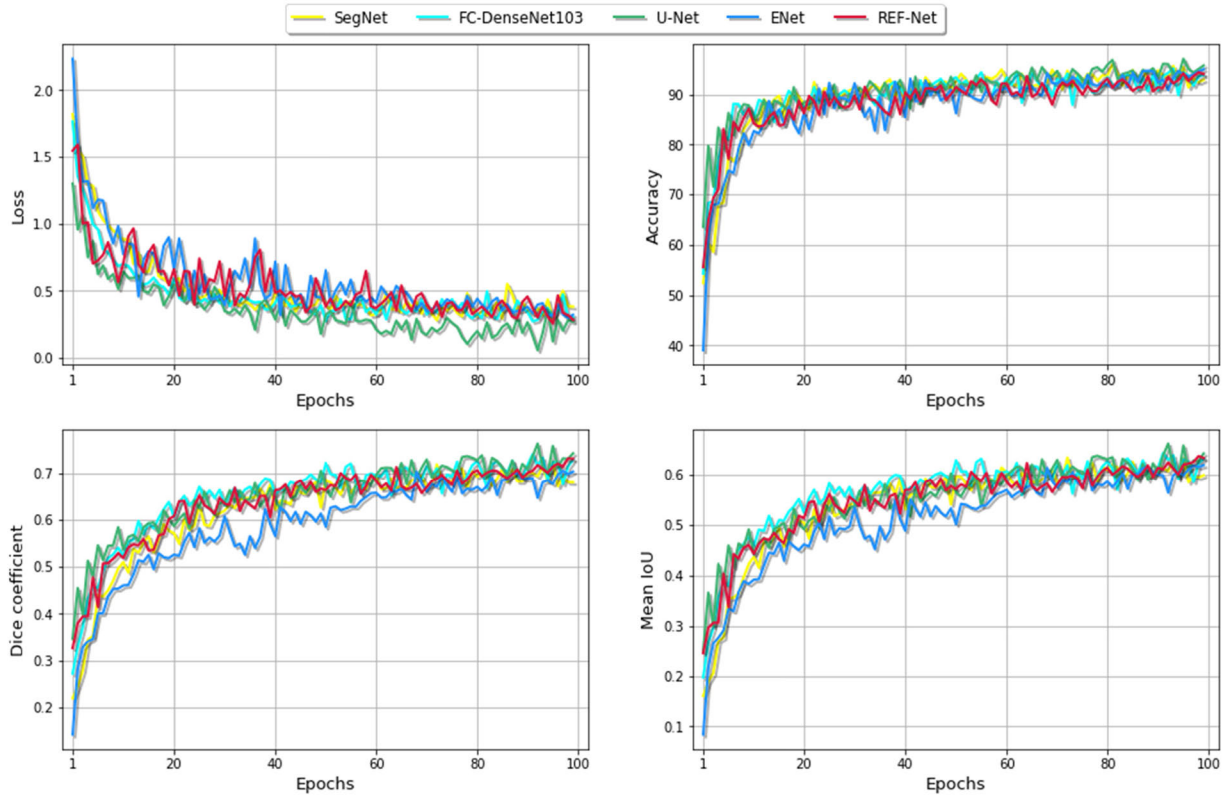

FIGURE 6. Experimental results on the validation set of Cityscapes dataset.

TABLE 2. Comparison of the models in terms of memory and parameters*.

Model Names	SegNet	FC-DenseNet103	U-Net	ENet	REF-Net
Forward Pass Size (MB)	1,573.5	4,037.6	1,137.1	527.8	**521.7**
Backward Pass Size (MB)	2,016.4	4,966.9	1,580.2	866.4	**834.6**
Trainable Parameters Size (MB)	107.33	35.56	118.42	1.34	**1.31**
Model Size (MB)	3,697.3	9,042.1	2,837.6	1,395.6	**1,356.6**
Number of Trainable Parameters (thousand)	29,460	9,322	31,044	350,6	**344,03**

* The information provided in this table was calculated for an image with size of 480 × 360 using 32 GB NVIDIA Tesla V100-SXM2 GPU.

to more powerful, slower, and computationally expensive models.

A similar tendency of evaluation metrics can be observed in the model's performance results on the Cityscapes dataset, provided in Figure 6. However, based on the results represented in this figure, we can see that the models performed slightly better in the Cityscapes dataset than the CamVid database. This superior performance was obtained due to a significantly larger number of training instances of Cityscapes in comparison with the CamVid. We can see that U-Net outperformed the other models in all evaluation metrics obtaining the lowest loss value and highest DC, meanIoU, and PA scores. Regarding the proposed method, it achieved very similar results in the considered evaluation metrics to the ones of U-Net. Still, the REF-Net model's performance was negligibly inferior to the best-performed model (U-Net), which is more storage-intensive and computationally expensive. Considering this, we can conclude that the proposed model achieved the most optimal tradeoff in terms of efficiency, speed, and accuracy.

V. DISCUSSION

The results obtained from the training and validation sets of the considered datasets provided promising results for the REF-Net model. Due to its efficiently formulated architecture, the proposed model required nearly a hundred times less memory and a million times fewer trainable parameters than the powerful models attaining state-of-the-art performance. Despite being highly efficient, the proposed model outperformed (SegNet, FC-DenseNet103) or at least obtained competitive results (U-Net) in terms of the evaluation metrics. However, the validation set cannot provide a realistic idea of a semantic segmentation model's strength since its examples partake in the model's training stage

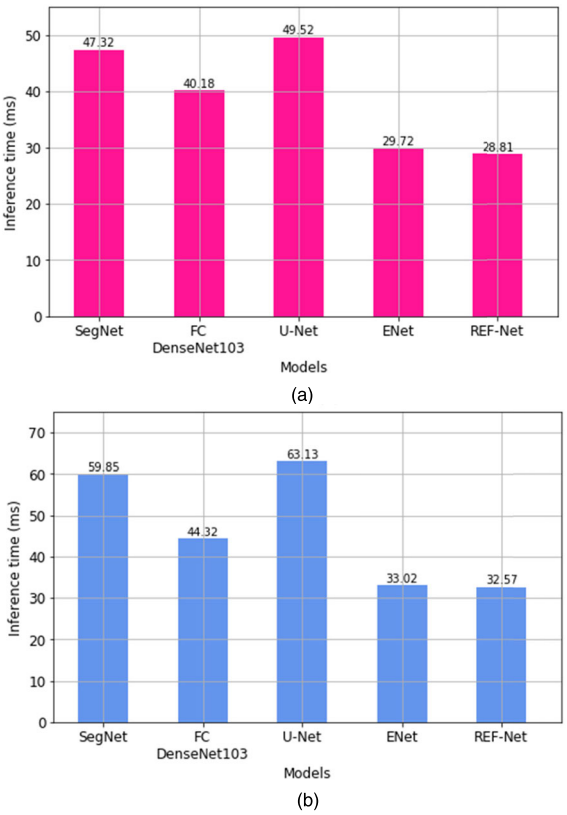

FIGURE 7. Required inference time to produce a segmented test image in (a) CamVid and (b) Cityscapes datasets.

and are used to fine-tune hyper-parameters. Therefore, the fairest comparison of the models could be obtained using a test set of the considered datasets. The test set was used only once after finishing training and acquiring the outcomes provided in this manuscript's experiments and results section.

A. DISCUSSION OF THE INFERENCE TIME

First, we discuss the considered models' required inference time. Figure 7 compares the baseline and proposed networks regarding the time required to generate a segmented test image. Similar to the training time results, REF-Net also performed considerably better in inference time. It required 28.81 and 32.57 milliseconds to output a segmented image from CamVid and Cityscapes test sets, respectively. These results were approximately two times faster than more powerful models, such as SegNet, FC DenseNet103, and U-Net that spent 47.32 and 59.85, 40.18 and 44.32, as well as 49.52 and 63.13 milliseconds to generate a segmented image in CamVid and Cityscapes datasets, respectively. Concerning an efficient ENet model, the proposed model performed nearly 3% faster than ENet on both datasets.

B. DISCUSSION OF THE GENERALIZIBILITY OF THE MODELS

We compared the considered models' ability to generalize by assessing their performance on the test sets using several evaluation metrics. The results of the baseline and proposed models on the considered datasets are provided in Table 3.

TABLE 3. The results of the baseline and proposed models on the test sets of the considered datasets.

Dataset Names	Evaluation metrics	SegNet	FC-DenseNet103	U-Net	ENet	REF-Net
CamVid	Loss	0.571	0.454	0.404	0.437	**0.397**
	PA (%)	88.27	89.18	90.81	89.71	**91.20**
	DC	0.628	0.642	0.651	0.637	**0.659**
	mIoU	0.574	0.582	**0.597**	0.579	0.586
Cityscapes	Loss	0.419	0.364	**0.352**	0.427	0.371
	PA (%)	0.917	0.929	**0.934**	0.915	0.926
	DC	0.664	0.683	0.681	0.661	**0.688**
	mIoU	0.589	0.607	0.603	0.585	**0.609**

Table 3 shows that REF-Net outperformed the baseline models in all evaluation metrics, except for mean IoU when assessed using the CamVid database. Notably, the performance of the proposed model was only slightly lower than U-Net's highest achieved result regarding the mean IoU metric. Considering that mean IoU equals to the fraction of true positives with regards to the sum of true positives, false positives, and false negatives, we can conclude that the proposed method produced more false positives and false negatives in comparison with U-Net model. However, when the models' performances were evaluated on the Cityscapes dataset, the REF-Net attained the highest performance only on the Dice coefficient and mean IOU. Considering the other evaluation metrics, the proposed model obtained the second-highest results in terms of loss and pixel accuracy respectively. Notably, the REF-Net model showed inferior performance only compared to a significantly powerful model, such as FC-DenseNet103.

C. DISCUSSION OF THE EXPERIMENTAL RESULTS

Considering the results of the conducted experiments on two open source datasets, we can see that ENet model and the proposed method has smaller differences in comparison to the more computationally expensive models, such as U-Net, FC-DenseNet-103, and SegNet. The reason for this is that both models (ENet and REF-Net) are considered as efficient models and tackle the problem of efficient computation in DCNNs. Specifically, ENet was originally proposed to deal with the problem of limited computational resources and excessive computation time in DCNNs. REF-Net, in turn, is also an efficient network that further improved the solutions to the aforementioned aspects of DCNN training; therefore, the proposed method achieved similar (better) results than ENet. Regarding the other computationally expensive models, their performance was significantly different because they were not regarded as efficient models. In short, for benchmarking, we selected ENet as an efficient model and U-Net, FC-DenseNet-103, and SegNet

FIGURE 8. (a) Input image, (b) target annotated image from the CamVid database, and (c) generated segmented image using the proposed model.

models as accurate models to compare their performance with the one of REF-Net. Because, the main objective of this research work was to propose both efficient and accurate model.

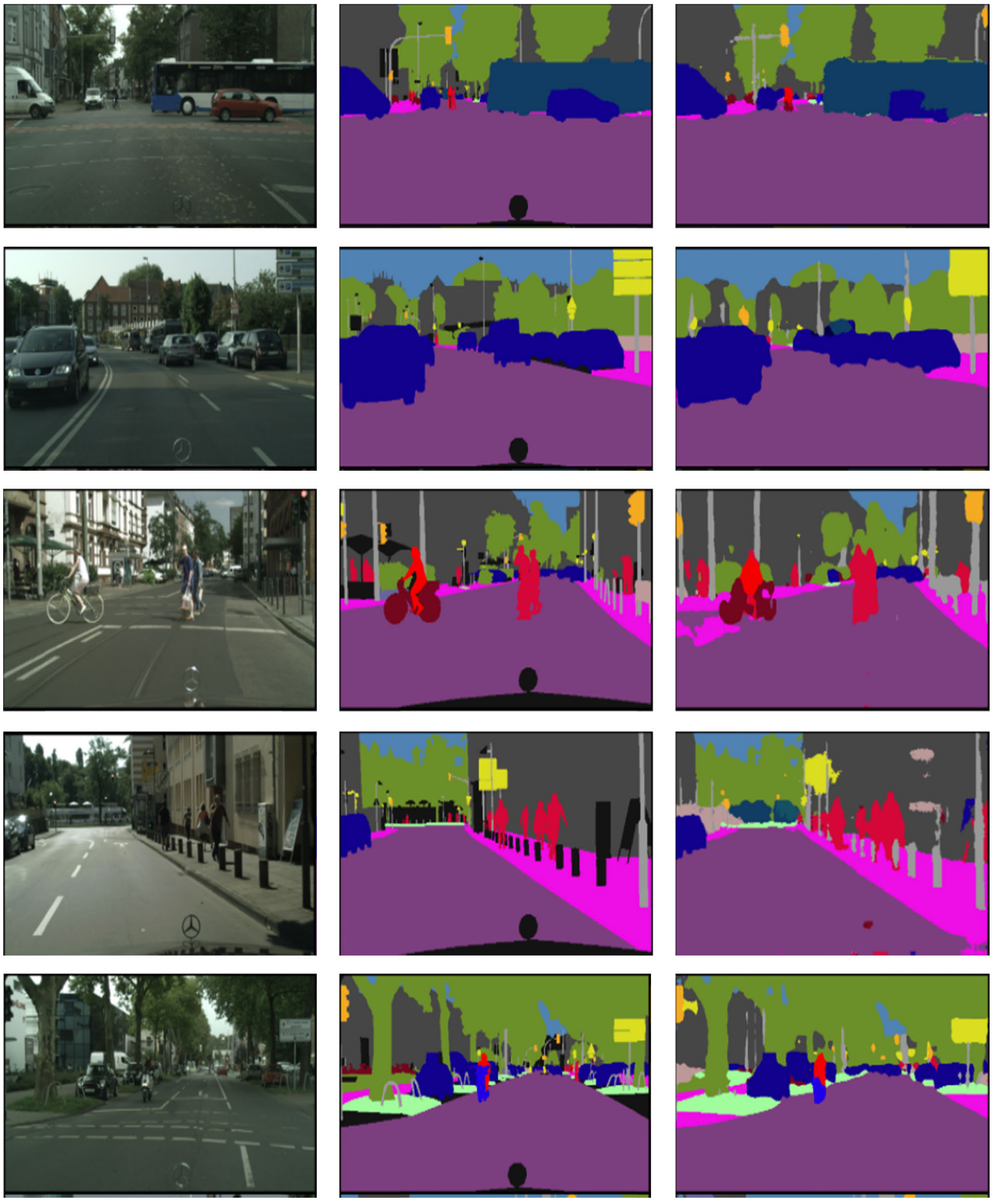

FIGURE 9. (a) Input image, (b) target annotated image from the Cityscapes dataset, and (c) generated segmented image using the proposed model.

D. DISCUSSION OF SEGMENTED IMAGES GENERATED BY THE PROPOSED MODEL

In addition to outperform its counterparts in several evaluation metrics on the test sets of the considered datasets and obtaining the best performance in efficiency and accuracy among the considered popular networks for semantic segmentation, the proposed method generated notable segmented images. These images on the test set of CamVid and Cityscapes datasets are represented in Figure 8 and Figure 9, respectively.

Based on the shown segmented images produced by REF-Net, we can see that they did not perfectly match with the ground truth masks. To the best of our knowledge, there was not any impeccable model that could produce identical segmented images to the target masks. However, the proposed model generated decent outputs that seem to be almost identical to the target annotated images. Considering this, we believe that the proposed model has a great potential of being effectively applied in developing software for mobile and battery-powered computational devices or real-time segmentation applications.

VI. CONCLUSION AND FUTURE WORK

In this study, we conducted research on semantic segmentation models in autonomous-driving. We also explored widely used DL models in this field that exhibited high processing complexity and enormous memory requirements, which did not allow the development of applications for devices with limited computational resources. Considering the increasing demand for mobile and battery-powered devices, we formulated the REF-Net model, which uses dilated and asymmetric convolution operations with skip connections and bottleneck layers in the contraction path. The nearest-neighbor interpolation-based upsampling method was also utilized to restore encoded images, requiring no trainable parameters at all.

In the experiments conducted with popular, publicly available datasets related to autonomous-driving, the REF-Net model required considerably fewer parameters, significantly less memory space, and substantially less training and inference time than the more powerful semantic segmentation models. Also, unlike the ENet model, REF-Net attained competitive accuracy results when assessed using several evaluation metrics. These facts ensured that the proposed model could be successfully implemented in applications with limited computational power with insignificant or no accuracy loss.

We plan to continue research work in autonomous-driving and develop more efficient and accurate DL model architecture by fine-tuning and enhancing the proposed model.

REFERENCES

[1] J. S. Sevak, A. D. Kapadia, J. B. Chavda, A. Shah, and M. Rahevar, "Survey on semantic image segmentation techniques," in *Proc. Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2017, pp. 306–313, doi: 10.1109/ISS1.2017.8389420.

[2] Z. Guo, X. Li, H. Huang, N. Guo, and Q. Li, "Deep learning-based image segmentation on multimodal medical imaging," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 3, no. 2, pp. 162–169, Mar. 2019, doi: 10.1109/trpms.2018.2890359.

[3] C. Lian, S. Ruan, T. Denoeux, H. Li, and P. Vera, "Joint tumor segmentation in PET-CT images using co-clustering and fusion based on belief functions," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 755–766, Feb. 2019, doi: 10.1109/TIP.2018.2872908.

[4] Y. Huo, Z. Xu, S. Bao, C. Bermudez, H. Moon, P. Parvathaneni, T. K. Moyo, M. R. Savona, A. Assad, R. G. Abramson, and B. A. Landman, "Splenomegaly segmentation on multi-modal MRI using deep convolutional networks," *IEEE Trans. Med. Imag.*, vol. 38, no. 5, pp. 1185–1196, May 2019, doi: 10.1109/TMI.2018.2881110.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Cham, Switzerland: Springer, 2015, doi: 10.1007/978-3-319-24574-4_28.

[6] Z. Zhao, L. Yang, H. Zheng, I. H. Guldner, S. Zhang, and D. Z. Chen, "Deep learning based instance segmentation in 3d biomedical images using weak annotation," in *Medical Image Computing and Computer Assisted Intervention* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-030-00937-3_41.

[7] H. Seo, M. B. Khuzani, V. Vasudevan, C. Huang, H. Ren, R. Xiao, X. Jia, and L. Xing, "Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications," *Med. Phys.*, vol. 47, no. 5, pp. e148–e167, 2020, doi: 10.1002/mp.13649.

[8] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Aug. 2019, pp. 5901–5904, doi: 10.1109/igarss.2019.8900532.

[9] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 574–586, Jan. 2019, doi: 10.1109/TGRS.2018.2858817.

[10] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The inria aerial image labeling benchmark," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3226–3229, doi: 10.1109/IGARSS.2017.8127684.

[11] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "DeepFashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5337–5345, doi: 10.1109/CVPR.2019.00548.

[12] M. Hou, L. Wu, E. Chen, Z. Li, V. W. Zheng, and Q. Liu, "Explainable fashion recommendation: A semantic attribute region guided approach," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1–8, doi: 10.24963/ijcai.2019/650.

[13] S. C. Hidayati, T. W. Goh, J.-S.-G. Chan, C.-C. Hsu, J. See, L.-K. Wong, K.-L. Hua, Y. Tsao, and W.-H. Cheng, "Dress with style: Learning style from joint deep embedding of clothing styles and body shapes," *IEEE Trans. Multimedia*, vol. 23, pp. 365–377, 2021, doi: 10.1109/tmm.2020.2980195.

[14] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, "MODNet: Motion and appearance based moving object detection network for autonomous driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2859–2864, doi: 10.1109/ITSC.2018.8569744.

[15] D. Feng, C. Haase-Schutz, L. Rosenbaum, H. Hertlein, C. Glaser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 17, 2020, doi: 10.1109/tits.2020.2972974.

[16] D. Zhou, J. Fang, X. Song, L. Liu, J. Yin, Y. Dai, H. Li, and R. Yang, "Joint 3D instance segmentation and object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1839–1849, doi: 10.1109/cvpr42600.2020.00191.

[17] A. Taeihagh and H. S. M. Lim, "Governing autonomous vehicles: Emerging responses for safety, liability, privacy, cybersecurity, and industry risks," *Transp. Rev.*, vol. 39, no. 1, pp. 103–128, Jan. 2019, doi: 10.1080/01441647.2018.1494640.

[18] J. Hu, P. Bhowmick, F. Arvin, A. Lanzon, and B. Lennox, "Cooperative control of heterogeneous connected vehicle platoons: An adaptive leader-following approach," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 977–984, Apr. 2020, doi: 10.1109/LRA.2020.2966412.

[19] Y. Huang and Y. Chen, "Autonomous driving with deep learning: A survey of state-of-art technologies," 2020, *arXiv:2006.06091*. [Online]. Available: http://arxiv.org/abs/2006.06091

[20] H. Ren, H. R. Karimi, R. Lu, and Y. Wu, "Synchronization of network systems via aperiodic sampled-data control with constant delay and application to unmanned ground vehicles," *IEEE Trans. Ind. Electron.*, vol. 67, no. 6, pp. 4980–4990, Jun. 2020, doi: 10.1109/TIE.2019.2928241.

[21] B. Major, D. Fontijne, A. Ansari, R. T. Sukhavasi, R. Gowaikar, M. Hamilton, S. Lee, S. Grzechnik, and S. Subramanian, "Vehicle detection with automotive radar using deep learning on range-azimuth-Doppler tensors," in *Proc. Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 924–932, doi: 10.1109/ICCVW.2019.00121.

[22] A. Danzer, T. Griebel, M. Bach, and K. Dietmayer, "2D car detection in radar data with PointNets," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 61–66, doi: 10.1109/ITSC.2019.8917000.

[23] R. Nabati and H. Qi, "RRPN: Radar region proposal network for object detection in autonomous vehicles," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3093–3097, doi: 10.1109/ICIP.2019.8803392.

[24] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499, doi: 10.1109/CVPR.2018.00472.

[25] Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, "RT3D: Real-time 3-D vehicle detection in LiDAR point cloud for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3434–3440, Oct. 2018, doi: 10.1109/LRA.2018.2852843.

[26] J. Beltran, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "BirdNet: A 3D object detection framework from LiDAR information," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3517–3523, doi: 10.1109/ITSC.2018.8569311.

[27] J. Zhou, X. Tan, Z. Shao, and L. Ma, "FVNet: 3D front-view proposal generation for real-time object detection from point clouds," in *Proc. 12th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2019, pp. 1–8, doi: 10.1109/CISP-BMEI48845.2019.8965844.

[28] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-FPN: Multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds," *Sensors*, vol. 20, no. 3, p. 704, Jan. 2020, doi: 10.3390/s20030704.

[29] C. Yan and E. Salman, "Mono3D: Open source cell library for monolithic 3-D integrated circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 3, pp. 1075–1085, Mar. 2018, doi: 10.1109/TCSI.2017.2768330.

[30] T. He and S. Soatto, "Mono3D++: Monocular 3D vehicle detection with two-scale 3d hypotheses and task priors," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8409–8416, doi: 10.1609/aaai.v33i01.33018409.

[31] G. Brazil and X. Liu, "M3D-RPN: Monocular 3D region proposal network for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9287–9296, doi: 10.1109/ICCV.2019.00938.

[32] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2040–2049, doi: 10.1109/CVPR.2017.198.

[33] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[34] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 11–19, doi: 10.1109/CVPRW.2017.156.

[35] X. Hu, M. A. Naiel, A. Wong, M. Lamm, and P. Fieguth, "RUNet: A robust UNet architecture for image super-resolution," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 505–507, doi: 10.1109/CVPRW.2019.00073.

[36] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "ResUNet—A: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS J. Photogramm. Remote Sens.*, vol. 162, pp. 94–114, Apr. 2020, doi: 10.1016/j.isprsjprs.2020.01.013.

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

[38] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587503.

[39] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Computer Vision—ECCV* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 2008, doi: 10.1007/978-3-540-88682-2-5.

[40] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr, "Combining appearance and structure from motion features for road scene understanding," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2009, pp. 138–147, doi: 10.5244/C.23.62.

[41] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 739–746, doi: 10.1109/ICCV.2009.5459248.

[42] P. Kohli, L. Ladický, and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency," *Int. J. Comput. Vis.*, vol. 82, no. 3, pp. 302–324, May 2009, doi: 10.1007/s11263-008-0202-0.

[43] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, Jan. 2009, doi: 10.1007/s11263-007-0109-1.

[44] X. He, R. S. Zemel, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2004, p. 2, doi: 10.1109/cvpr.2004.1315232.

[45] A. Dewan, G. L. Oliveira, and W. Burgard, "Deep semantic classification for 3D LiDAR data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 3544–3549, doi: 10.1109/IROS.2017.8206198.

[46] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast LIDAR-based road detection using fully convolutional neural networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1019–1024, doi: 10.1109/IVS.2017.7995848.

[47] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2626–2635, doi: 10.1109/CVPR.2018.00278.

[48] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440, doi: 10.1109/CVPR.2015.7298965.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[50] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708, doi: 10.1109/CVPR.2017.243.

[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1–11.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034, doi: 10.1109/ICCV.2015.123.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[56] C. Szegedy *et al.*, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. Aritifical Intell.*, 2017, vol. 31, no. 1, doi: 10.1089/pop.2014.0089.

[57] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–13.

[58] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet v2: Bilateral network with guided aggregation for real-time semantic segmentation," 2020, *arXiv:2004.02147*. [Online]. Available: http://arxiv.org/abs/2004.02147

[59] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

BEKHZOD OLIMOV received the B.S. degree in economics from the Fergana Polytechnic Institute, Uzbekistan, in 2014, and the M.S. degree from Yeungnam University, South Korea, in 2018. He is currently pursuing the Ph.D. degree with the Computer Science and Engineering Department, Kyungpook National University, South Korea. His research interests include computer vision and pattern recognition using deep learning techniques.

JEONGHONG KIM received the B.S. and M.S. degrees from Kyungpook National University, Daegu, South Korea, in 1986, and the Ph.D. degree from Chungnam National University, Daejeon, South Korea, in 2001.

He worked as a Senior Researcher with the Electronics and Telecommunications Research Institute from 1988 to 1996. He also worked as a Professor with Sangju National University from 1996 to 2008. He is currently working as a Professor with the School of Computer Science and Engineering, Kyungpook National University. His current research interests include bio signal processing and pattern recognition using deep learning techniques.

ANAND PAUL (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from National Cheng Kung University, Taiwan, R.O.C., in 2010.

He is currently an Associate Professor with the School of Computer Science and Engineering, Kyungpook National University, South Korea. His research interests include algorithm and architecture reconfigurable embedded computing. He serves as a Reviewer for various journals such as IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, IEEE SENSORS, *ACM Transactions on Embedded Computing Systems*, *IET Image Processing*, *IET Signal Processing*, and *IET Circuits and Systems*.