IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 September 2023, accepted 22 October 2023, date of publication 25 October 2023, date of current version 1 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327323

RESEARCH ARTICLE

SDDS-Net: Space and Depth Encoder-Decoder Convolutional Neural Networks for Real-Time Semantic Segmentation

HATEM IBRAHEM[1], AHMED SALEM[1,2], AND HYUN-SOO KANG[1], (Member, IEEE)
[1]School of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, South Korea
[2]Department of Electrical Engineering, Faculty of Engineering, Assiut University, Assiut 71515, Egypt

Corresponding author: Hyun-Soo Kang (hskang@cbnu.ac.kr)

This work was supported in part by the Chungbuk National University BK21 Program, in 2023; and in part by the Research Projects of ''Development of Automatic Screening and Hybrid Detection System for Hazardous Material Detecting in Port Container'' funded by the Ministry of Oceans and Fisheries under Grant 20200611.

ABSTRACT In this paper, we propose novel convolutional encoder-decoder architectures for real-time semantic segmentation based on an image-to-image translation approach via the space-to-depth and depth-to-space modules. We present architectures that compress the spatial information of the image using the space-to-depth (SD) instead of the commonly used pooling methods (Max-pooling and Average-pooling) or strided convolution approaches. The SD module can reduce the image size while preserving the spatial information of the image in the form of extra depth information, this approach is much better than the pooling approaches which introduce a loss in the information and the details of the image. We also propose a lightweight and simple decoder stage using the depth-to-space (DS) module which constructs a high-resolution dense prediction map from a large number of low-resolution feature maps. The proposed architectures are efficient in learning image classification and semantic segmentation with high accuracy and average processing speed. We trained and tested our proposed architectures on image classification (i.e. CIFAR10 and Tiny ImageNet), and indoor and outdoor benchmarks for semantic segmentation specifically NYU-depthV2 and CITYSCAPES. The proposed architectures could attain high accuracy in classification (94.28% on CIFAR10 and 72.25% on Tiny ImageNet) and high mean average precision and pixel accuracy values in semantic segmentation (pixel accuracy of 78.55% on NYU-depthV2 and 87.9% on CITYSCAPES) while maintaining a real-time speed of frame processing outperforming recent state-of-the-art methods in semantic segmentation.

INDEX TERMS Convolutional neural networks, image classification, image-to-image translation, real-time processing, semantic segmentation.

I. INTRODUCTION

The general architecture of the convolutional neural networks (CNN) uses a down-sampling method (e.g., pooling or strides in convolution layers) to compress the representation to a more informative one, make the training process more efficient, and speed up the training process. Most of the CNN architectures use Max-pooling (MP) or Average pooling (AP) to compress the feature space to more reduced representation, however, MP and AP introduce information loss as they compress the features considering that the important information exists only in the maximum value or the average value of the window used to slide over the input data. Those pooling approaches give a lossy compressed representation which negatively affects the overall learning process using the neural network architecture due to the information lost during the pooling process. Other researches [1], [2], [3] showed that the strided convolution in some architectures is able to learn

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

the best down-sampling parameters and is better than the Max-pooling which is a non-learnable mathematical process however the strided convolution adds more parameters and complexity to the model compared to the pooling dependent models. The recent progress in CNN architectures has shown the superior ability of CNNs in performing many computer vision tasks however most CNN models use inefficient feature compression methods. Among the recent critical tasks in computer vision, semantic segmentation is one of those important tasks. It is employed in robotics [4], 3D image understanding [5], medical diagnosis [6], Virtual/Augmented reality [7], Video coding (Region-of-interest coding) [8], [9], and self-driving vehicles [10]. Thus performing this task in real-time is extremely beneficial for those applications. Many models have achieved challenging semantic segmentation performance depending on CNN architectures. Some researches [13], [14] showed that a single architecture can perform multiple computer vision tasks. Almost all the existing semantic segmentation architectures use MP or strided convolution for feature compression, which are inefficient as they introduce a loss of information. We address the problem of the optimized method for the down-sampling of the features without losing major or minor information. The proposed down-sampling method reduces the spatial size of the input features however adds the spatial reduction as extra depth channels through a convolutional learnable technique that preserves the same amount of information. The proposed method uses the space-to-depth (SD) module [11] and the depth-to-space (DS) module [12] which were originally proposed for the image/video super-resolution task. Our proposed method, namely the Space-to-Depth encoder and Depth-to-Space decoder network (SDDS-Net) can perform the task of semantic segmentation with high accuracy. We can brief our contribution in this paper as follows:

- We propose new convolutional encoder-decoder architectures based on the robust SD and DS learnable modules which can learn dense prediction (semantic segmentation) task efficiently.
- We compare the performance of different encoder architecture configurations such as convolutional architecture with MP, convolutional architecture with strided convolution, convolutional architecture with SD, and depth-wise separable convolutional architecture with residual connections and SD.
- We show that our proposed method can perform semantic segmentation with high speed of processing (∼25).

We first experiment with the SD downsampling-based architecture in the task of the image classification to prove the robustness of the SD module in the image and feature downsampling and its performance compared to the traditional Max-pooling and strided convolution. Then, we extend the classification architecture to perform semantic segmentation based on an image-to-image translation approach. The organization of the remaining of this paper is as follows, section II presents the related work to our proposed work, section III presents the details of the proposed method, section IV presents the experiments done in this work, section V presents the results obtained by the proposed methods and comparisons with other state-of-the-art (SOTA) methods in semantic segmentation, and section VI states the conclusion of our work.

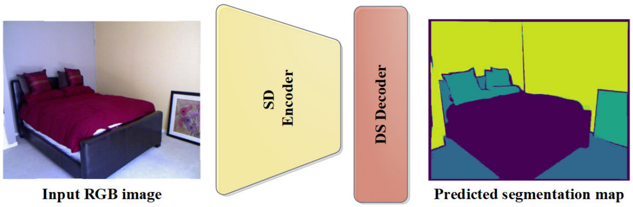

FIGURE 1. The general architecture of the proposed method with Space-to-Depth (SD) encoder and Depth-to-Space (DS) decoders for the task of semantic segmentation.

II. RELATED WORK

The two key ideas behind our architecture are the SD layer [11] which was proposed by Wang et al. to down-sample a high-resolution optical flow map to a low-resolution map with extra depth channel for video super-resolution task, and the DS layer [12] which was proposed by Shi et al. under the name ''efficient sub-pixel CNN'', they used this layer to construct a high-resolution image from many low-resolution images. The SD layer is employed in the encoder stage as a down-sampling module similar to the pooling methods with an output depth dependent on the spatial size of the input data. While the DS module is used as the decoder stage to construct the high-resolution dense prediction map from the small feature maps generated by the encoder stage.

In Image classification, the recent convolutional neural networks have shown an outstanding performance in the task of image classification, especially ImageNet classification. Most of those architectures [15], [19], [20], [21], [22], [23], [30], [37], [49] are based on Maxpooling for the downsampling of the images or features. Other research such as Inception [16] presented a hybrid approach of downsampling using both strided convolution with different kernel sizes and Max-pooling, then the output of all operations is concatenated. Springenberg et al. [3] proposed the all-convolution network which depends exclusively on the strided convolution for down-sampling. Xie et al. [24] proposed ResNext which depends mainly on the strided 3 × 3 convolution with a stride of 2 for downsampling. Liu et al. [25] proposed ConvNext which also depends on strided convolution in addition to an image patching approach instead of the whole image as an input. Although the previous methods are efficient in learning the image classification task, it also introduce some information loss due to the dependency on inefficient downsampling techniques. The proposed downsampling approach using convolution and SD module grantee the largest possible feature information compared to the max-pooling and the strided convolution.

Semantic segmentation is that dense prediction task that aims to predict the label of each pixel in an image. Most of the recent research on semantic segmentation [28], [29], [31], [32], [35], [36] employ encoder-decoder CNN architectures to perform such task. Fully convolutional networks (FCN) [29] was the first encoder-decoder architecture that used VGG16 [30] classification network for segmentation after removing the few final fully connected layers and added an up-sampling layer as a decoder. SegNet [31] and U-Net [32] are the most popular encoder-decoder architectures which employed encoding architectures to compress the input image to a latent vector then they constructed the semantic segmentation predictions using a deconvolution decoder stage with some other tricks such as pooling location sharing between encoder and decoder in SegNet and residual connections between the encoder and decoder layers in U-Net. Chen et al. [33], [34], [35], [36] proposed four versions of their approach 'Deeplab' which aimed to perform semantic segmentation efficiently. In DeeplabV1 [33], the authors proposed the Atrous algorithm to increase the receptive field of the convolution and they also proposed Conditional Random Field (CRF) to enable the model to learn the fine details of the objects in the image. In DeeplabV2 [34], they proposed the multi-scale processing using the Atrous Spatial Pyramid Pooling (ASPP) and they replaced VGG16 architecture with ResNet101 [37]. While in DeeplabV3 [35], they improved the ASPP by adding different sampling rates and batch normalization layers, they also showed that using 1 × 1 convolution is better 3 × 3 to eliminate the image boundary effect. Finally, in Deeplabv3+ [36], they proposed a depth-wise separable convolutional encoder using Aligned-Xception [38] and they optimized the decoder stage to improve the accuracy of the segmentation learning process. The recent state-of-the-art methods on semantic segmentation use transformers to model that task, transformers deal with the image in a similar way to the text, in which there is an inter-dependency between the words in a phrase. The transformer deals with the image as a sequence of patches where there are inter-relations between those patches. Zheng et al. [39] proposed a sequence-to-sequence transformers-based method to perform the semantic segmentation task using an image patch encoder to model the global context of the input image and employed a simple decoder to provide the segmentation. While Wu et al. [40] proposed fully transformer networks (FTN) for semantic segmentation using a pyramid group transformer as a convolutional transformer encoder. All the previously mentioned methods employ a complex implementation of the encoder and decoder networks, while the proposed method employs a simple encoder and decoder implementation however it outperforms the SOTA methods in semantic segmentation. The DS module is proved to be superior in feature decoding in dense prediction tasks (semantic segmentation and depth estimation) as it is applied in recent research [41], [42], [43].

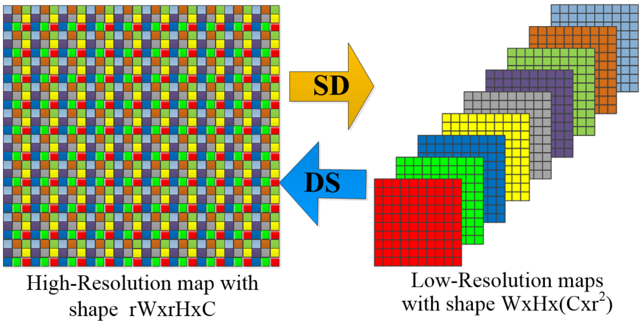
SD
DS
High-Resolution map with shape rWxrHxC
Low-Resolution maps with shape WxHx(Cxr^2)

FIGURE 2. The two main modules in our proposed method. a) Space-to-Depth (SD) which is used to down-sample the input feature map of size $rW \times rH \times C$ to a lower resolution map of size $W \times H \times C \times r^2$ via a learnable process. b) Depth-to-Space (DS) which is used as the decoder stage in our method to up-sample the input low-resolution feature map of size $W \times H \times r^2$ to a higher resolution map of size $rW \times rH$ through a learnable process.

III. PROPOSED METHOD

The proposed method depends on two main blocks; the SD layer as a down-sampling module similar to the pooling layers, and the DS layer is used as the decoder stage to merge the feature depth in order to up-sample the feature maps to form the dense map at the same size of the input.

A. SPACE-TO-DEPTH AS AN ENCODING LAYER

Space-to-Depth (SD) module was first proposed by Wang et al. [11] as a way of obtaining a dense representation of the optical flow to be used for video super-resolution. In our proposed method, we employ it as a learnable spatial down-sampling layer similar to the pooling method. The difference in the SD module from pooling is that no feature compression happens to the input feature maps but the reduction in the spatial size is converted to depth data via pixel aggregation technique. This pixel aggregation is done by converting input feature maps of shape $rW \times rH \times C$ into feature maps of shape $W \times H \times C \times r^2$ through a learnable aggregation process which mathematically can be stated as follows:

$$Y^{W \times H \times C \times r^2} = W_L * f^{L-1}(X^{rW \times rH \times C}) + b_L, \qquad (1)$$

where Y and X are the low-resolution output with extended depth channel and the high-resolution input of the DS layer, respectively. W_L and b_L are the weights and biases in the DS layer, W is the image width, H is the image height, C is the image channels, r is the depth of the image, and f is the activation function for the layer. This layer is applied five times in our proposed architectures each time it reduces the spatial size by r=2 in the width and r=2 in the height and increases the depth 4 times ($r^2 = 4$). Each time the input image is down-sampled, convolutional layers, relu, and batch normalization are applied in a different order depending on the architecture.

B. DEPTH-TO-SPACE AS A DECODER NETWORK

Depth-to-space (DS) module was first proposed by Shi et al. [12] as a way of aggregating the pixels of the input features to obtain a higher-resolution image for the image

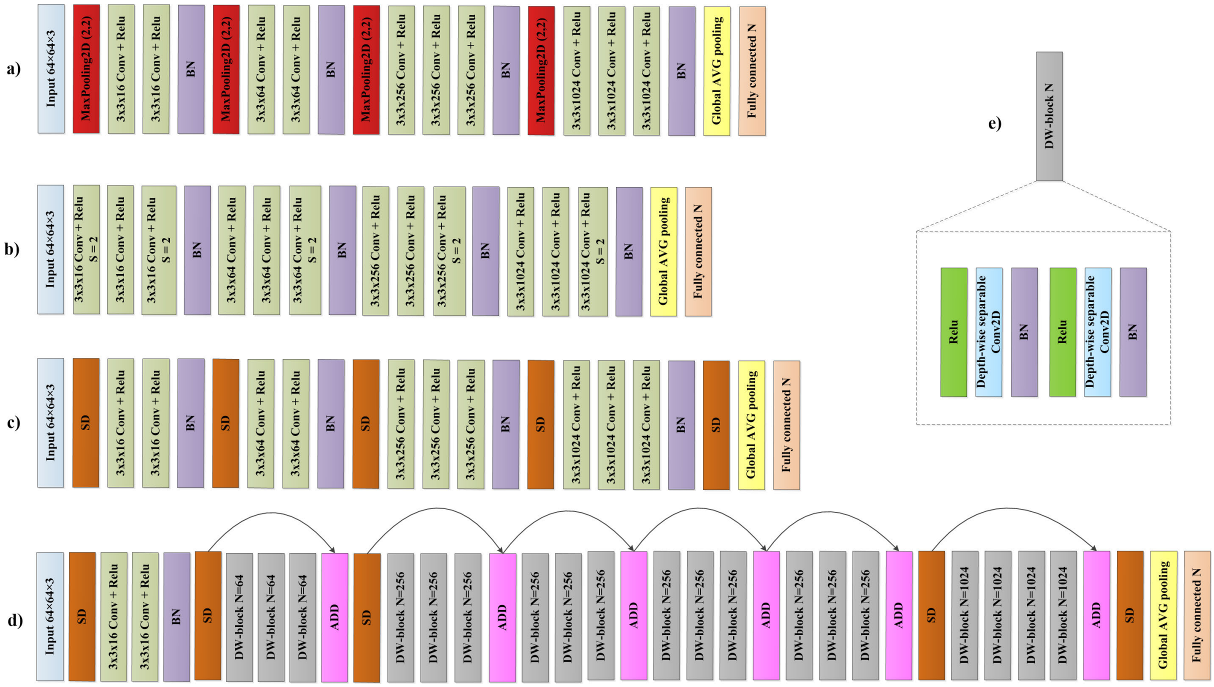

FIGURE 3. The proposed architectures for image classification: (a) CNN architecture with max-pooling (MP) for down-sampling, (b) CNN architecture with strided convolution for down-sampling (s refers to both the vertical and the horizontal strides), (c) CNN architecture with Depth-to-space (SD) for down-sampling, d) CNN with depthwise separable convolution and SD for down-sampling, and e) is the depthwise block (DW-block) (used in the architecture d) which consists of a single repetition of Relu followed by depth-wise separable convolution and batch normalization.

super-resolution task. In our proposed method, we employ it as a one-stage learnable up-sampling decoder. This pixel aggregation is done by converting the input feature maps of shape $W \times H \times r^2$ obtained from the encoder into a dense map of shape $rW \times rH$ through a learnable process which mathematically can be stated as follows:

$$Y^{rW \times rH} = W_L * f^{L-1}(X^{W \times H \times r^2}) + b_L, \tag{2}$$

This layer is applied five times in our proposed architectures as the decoding stage, it up-samples the image of the final feature maps obtained from the encoder stage by a factor of 32 (2^5) to obtain a dense prediction map at the same size of the input image.

C. PROPOSED ARCHITECTURES

We propose four architectures with different CNN configurations and we compare their performance and highlight the advantages of each one. We propose a simple CNN applying max-pooling to reduce the spatial size of the input features with repeated two or three convolutional layers with Relu activation followed by batch normalization (BN). The feature depth through the down-sampling stages are 3, 16, 64, 256, and 1024 and then a global average pooling followed by a fully connected layer is added in case of image classification. In the case of semantic segmentation, the final dense map is constructed from 1024 low-resolution features obtained by 1×1 convolutional layer at a size of $32 \times W$ and $32 \times H$ using the DS decoder, the first convolutional architecture is shown in Figure 3-a. The second architecture is a CNN architecture with the same configurations as the previous one but by replacing the max-pooling with a 3×3 strided convolution, we remove the final convolution of each block of the three final blocks and modify the stride to be 2 in the final convolution each block as shown in figure 3-b. The third CNN architecture is SD-Net (SDDS-Net for segmentation) which also has the same architecture as the first architecture with MP but the SD layer is applied instead of MP to down-sample the spatial size of the input and extend the depth of the output features as shown in figure 3-c.

The fourth proposed architecture is an architecture with depth-wise separable convolution (DW) [49] and residual connections so-called SD-Net-DW (SDDS-Net-DW for segmentation). The depth-wise separable convolution is another type of convolution proposed by François [49] which consists of depth-wise convolution (convolution for each channel separately) and point-wise convolution (1×1 convolution to project the depth of the features into less number of channels), DW-convolution is much faster than normal convolution as it has a lower number of parameter, exactly $\frac{1}{D} + \frac{1}{N}$ than that for conventional convolution as D and N are the size and the number of the input filter sequentially. We built this architecture using depth-wise block (DW-block) which consists of repeated Relu+dw-convolution+BN as shown in figure 3-e, gradually decreasing the spatial resolution and increasing the number of filters using the SD as shown in figure 3-d. Similar to other architectures, a global average pooling followed by a fully connected layer is added at the

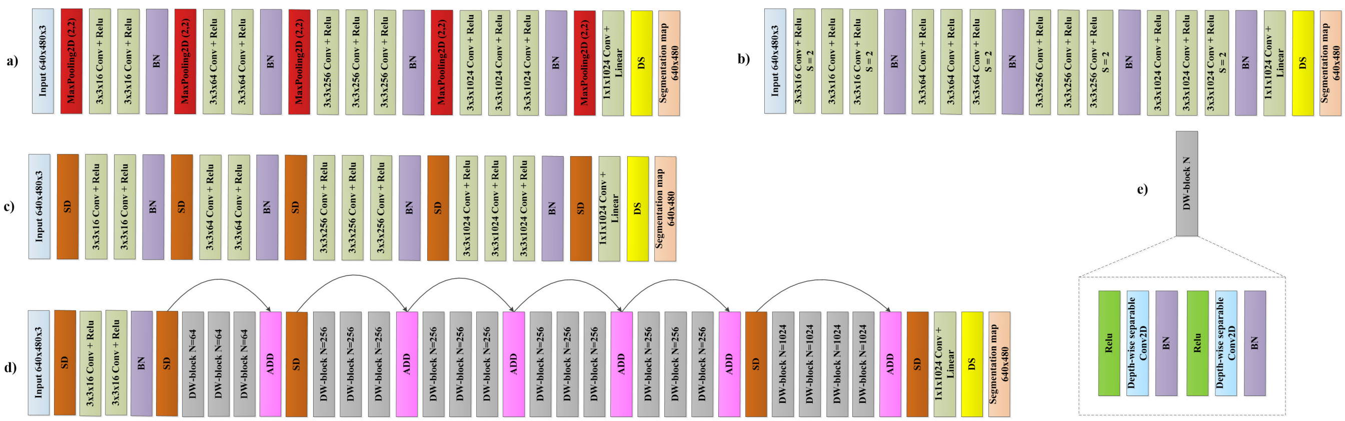

FIGURE 4. The proposed architectures for semantic segmentation: (a) CNN architecture with max-pooling (MP) for down-sampling and DS decoder, (b) CNN architecture with strided convolution for down-sampling (s refers to both the vertical and the horizontal strides) and DS decoder, (c) CNN architecture with Depth-to-space (SD) for down-sampling and DS decoder, d) CNN with depthwise separable convolution, SD for down-sampling, and DS decoder, and e) is the depthwise block (DW-block) (used in the architecture d) which consists of a single repetition of Relu followed by depth-wise separable convolution and batch normalization.

end of the architecture in case of classification. The final features are fed to 1 × 1 × 1024 to construct a dense map at the same size as the input image using the DS decoder in case of semantic segmentation. We compare the performance of the four proposed architectures in section VI (Results) showing that SDDS-Net and SDDS-Net-DW have much better accuracy than the MP-CNN and Strided-CNN. The proposed architectures for image classification and semantic segmentation are shown in Figure 3 and Figure 4, respectively where the difference is that in the case of semantic segmentation, the decoder network (1 × 1 convolution followed by a depth-to-space layer) is added instead of the global average pooling and the fully connected layer in case of image classification.

D. LOSS FUNCTION

The loss function used for image classification is the categorical cross-entropy loss as follows:

$$L_{CE} = -\sum_{i=1}^{N} q_i log(p_i) \tag{3}$$

where q is the ground truth label and p is the predicted label. i is an iterator over classes. The loss function used for learning the semantic segmentation is the Huber loss (a function which selectively operates either like L1 loss or L2 loss depending on a threshold value "t"), it is mathematically stated as:

$$L_{seg} = \begin{cases} \frac{1}{2r^2HW}\sum_{x=1}^{W}\sum_{y=1}^{H}(I_{x,y} - \tilde{I}_{x,y})^2, \text{if } |I - \tilde{I}| < t \\ \\ \frac{t}{r^2HW}\sum_{x=1}^{W}\sum_{y=1}^{H}(|I_{x,y} - \tilde{I}_{x,y}| - \frac{1}{2}t), \text{otherwise} \end{cases} \tag{4}$$

where I is the ground truth pixel value and $\tilde{I}$ is the predicted pixel value, while the threshold value, t, is selected as 1 since empirically it speeds up the training process. L1 and L2 are also tested separately for the proposed method training in two different experiments however each one had a slow loss improvement problem at some point during the training.

IV. EXPERIMENTS

We trained and tested our proposed method on image classification and semantic segmentation. For image classification, we trained and evaluated the proposed method on CIFAR10 and Tiny-ImageNet benchmarks. For semantic segmentation, we trained and evaluated the proposed models on the challenging NYU depth V2 and CITYSCAPES benchmarks to test the performance of the model on both indoor and outdoor scenes.

A. BENCHMARKS FOR IMAGE CLASSIFICATION EVALUATION

To evaluate the proposed encoding architectures using MP, strided convolution, and SD, we train and test the architectures on CIFAR10 [44] and Tiny ImageNet [45]. The CIFAR-10 is general scenes dataset consisting of 60,000 color images with 32 × 32 size divided into 10 classes with 6,000 images per class. There are 50,000 training samples and 10,000 test samples. Tiny ImageNet contains 200 classes of general image categories with a total of 100,000 images (500 for each class) of 64 × 64 size. Each class has 500 training samples, 50 validation samples, and 50 test samples.

B. BENCHMARKS FOR SEMANTIC SEGMENTATION EVALUATION

The first benchmark for semantic segmentation evaluation that we trained our models on is **NYU depth V2** [50]. It consists of 1,449 labeled images and 407,024 unlabeled images of indoor scenes (bedrooms, living rooms, kitchens, bathrooms, and offices) with their corresponding dense semantic segmentation map captured by Microsoft Kinect sensor. The dataset has 14 different pixel-level labels for

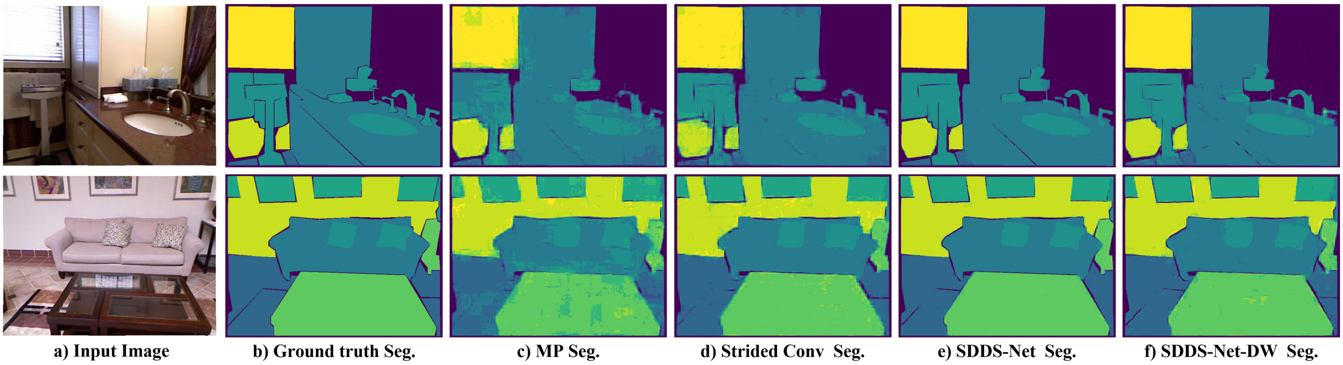

FIGURE 5. Comparison between the outputs obtained from each architecture: (a) Input RGB image. (b) The ground truth semantic segmentation mask. (c), (d), (e), and (f) are the predicted semantic segmentation masks from the following configurations: max-pooling (MP) based CNN architecture, the strided convolution-based CNN architecture, SDDS-Net (space-to-depth (SD) based CNN Architecture), and SDDS-Net-DW (SD and DW based architecture), respectively.

segmentation and the dataset is divided into 795 training images and 654 test images as we train our model on the labeled images only. The RGB images and the corresponding segmentation maps have the size of 640 × 480. We train our model on the same size without down-sampling. The model compresses the features in five stages to obtain feature maps of size 20 × 15 × 1024 which are used by the decoder to obtain the predicted dense map at the same input size $((20 \times 32) \times (15 \times 32) = 640 \times 480)$ as $r = \sqrt{1024} = 32$.

The second benchmark for semantic segmentation evaluation which we trained our model on is **CITYSCAPES** [51]. It consists of urban street scenes in Germany with their corresponding semantic segmentation maps with 19 different categories. The dataset contains 5000 fine-labeled images and 20,000 coarse-labeled images for semantic segmentation. We train and test our model on the fine-labeled images only since we aim to predict fine and clear segmentation maps. The RGB image size and the corresponding segmentation have the size of 2048 × 1024, we down-sample the images to 1024 × 512 to speed up the training process while keeping high-resolution predictions. The final dense prediction map is constructed at the same size as the input image using feature maps of the size 32 × 16 × 1024.

C. COMPARISON BETWEEN DIFFERENT CNN ARCHITECTURES

We compare our proposed architectures (SDDS-Net, SDDS-Net-DW) with the other architectures with similar configurations while using MP and strided convolution for down-sampling instead of the SD layer. The proposed MP-based CNN architecture is similar to SDDS-Net architecture but with replacing the SD module for down-sampling with MP as shown in Figure 3-a. While the strided convolution-based architecture has similar architecture but we replace the SD module with 3 × 3 convolution with strides of 2 in both horizontal and vertical axes as shown in Figure 3-b. In the result section, we show that our proposed architecture with DS for down-sampling attains much better accuracy in dense predictions than those using MP and strided convolution.

D. TRAINING AND TEST CONFIGURATIONS

We train and test our model on a desktop computer using the same hardware configuration. The hardware configuration includes Nvidia RTX3090 GPU which has Ampere RTX architecture and 24 GB of high-speed G6X memory, Intel Core i7-8700 CPU with 3.20 GHz clock speed, and 64 GB RAM. All the proposed architectures trained using Tensorflow Keras environment for 1500 epochs with Adam's optimizer with the standard image/mask augmentation, the training and test input image sizes for CIFAR10 and Tiny ImageNet are 32 × 32 and 64 × 64, respectively. The image size for NYU depth V2 is 640 × 480 and for CITYSCAPES is 1024 × 512.

V. RESULTS

In this section, we show the results obtained using the proposed method on CIFAR10, Tiny ImageNet, NYU depthV2, and CITYSCAPES benchmarks and we compare the obtained results with SOTA methods in image classification and semantic segmentation.

A. EVALUATION METRICS

We evaluate the classification performance using the classification accuracy (Acc.) using the following equation.

$$Acc. = \frac{1}{N}\sum_{c=1}^{N}\frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \tag{5}$$

where N is the number of classes, true-positive (TP) is the pixels that are truly predicted, true-negative (TN) is the pixels that are predicted that it is not of that class, false-positive (FP) is the pixels which are mispredicted to be that class, and false-negative (FN) is the pixels which are mispredicted to be not of that class.

We evaluate the semantic segmentation task using the mean intersection over union (mIOU) which is the area of intersection between the predicted P mask and the ground truth G mask over the union of the two masks as shown in

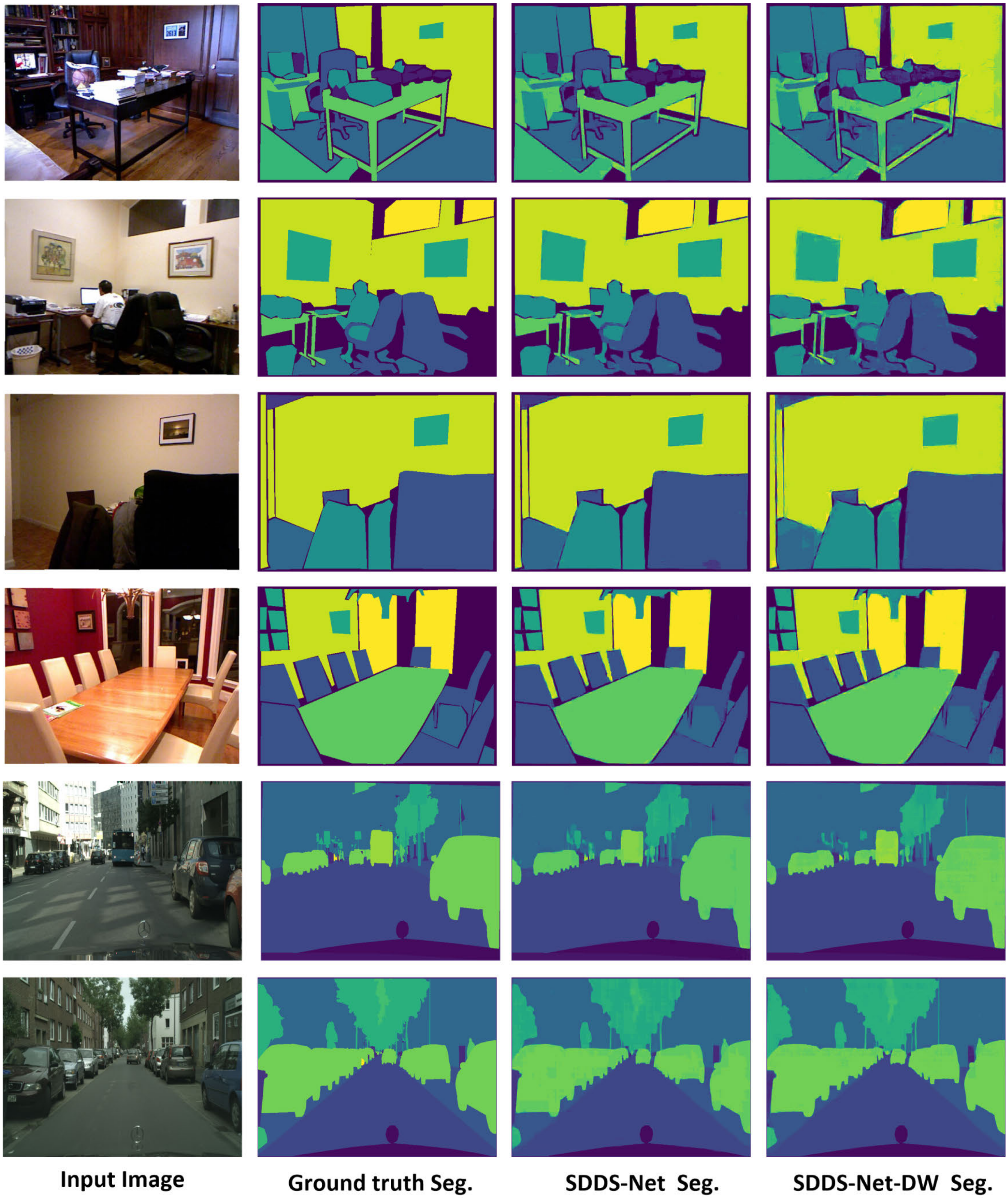

FIGURE 6. Sample results were obtained from the proposed architectures (SDDS-Net and SDDS-Net-DW) based on our method. The columns from left to right represent the input image, ground truth segmentation map, SDDS-Net predicted segmentation map, and SDDS-Net-DW predicted segmentation map. The first to fourth row represent samples from the NYU depthV2 test dataset and the fifth and sixth rows represent samples from the CITYSCAPES test dataset.

TABLE 1. Comparison between the obtained accuracy between the proposed architectures (MP-based CNN, Strided Conv. based CNN, SDDS-Net, and SDDS-Net-DW) on both CIFAR10 and Tiny ImageNet benchmarks reporting the model's parameters count (PC) in Millions for each model. Note that Strategy 1 is Mixup+Cutout and Strategy 2 is Randaug+Cutout.

Model	Dataset	PC	Aug.	Acc.
MP-CNN	CIFAR10	2.88	Strategy1	65.25
MP-CNN	CIFAR10	2.88	Strategy2	73.64
Strided-CNN	CIFAR10	2.95	Strategy1	71.55
Strided-CNN	CIFAR10	2.95	Strategy2	76.41
SD-Net	CIFAR10	3.08	Strategy1	90.40
SD-Net	CIFAR10	3.08	Strategy2	**94.28**
SD-Net-DW	CIFAR10	3.06	Strategy1	88.25
SD-Net-DW	CIFAR10	3.06	Strategy2	91.81
MP-CNN	T-ImageNet	3.03	Strategy1	57.54
MP-CNN	T-ImageNet	3.03	Strategy2	62.63
Strided-CNN	T-ImageNet	3.11	Strategy1	60.23
Strided-CNN	T-ImageNet	3.11	Strategy2	63.74
SD-Net	T-ImageNet	3.28	Strategy1	69.10
SD-Net	T-ImageNet	3.28	Strategy2	71.49
SD-Net-DW	T-ImageNet	3.26	Strategy1	64.91
SD-Net-DW	T-ImageNet	3.26	Strategy2	**72.25**

TABLE 2. Comparison between the obtained mIOU and speed in fps between the proposed architectures (MP-based CNN, Strided Conv. based CNN, SDDS-Net, and SDDS-Net-DW) on both NYU depthv2 and Cityscapes benchmarks reporting the model's parameters count (PC) in Millions for each model.

Model	Dataset	PC	Pix acc.	mIOU	fps
MP-CNN	NYU V2	23.66	74.15	60.11	26.3
Strided-CNN	NYUV2	23.70	74.06	63.23	25.4
SDDS-Net	NYUV2	34.36	**78.55**	**65.23**	25.0
SDDS-Net-DW	NYUV2	14.39	75.26	64.75	22.2
MP-CNN	CS	23.66	67.41	74.65	13.7
Strided-CNN	CS	23.70	70.36	75.89	13.0
SDDS-Net	CS	34.36	85.01	80.12	12.5
SDDS-Net-DW	CS	14.39	**87.9**	**82.35**	11.6

the following equation:

$$IOU = \frac{1}{N} \sum_{c=1}^{N} \frac{P \cap G}{P \cup G} \tag{6}$$

we also evaluate the segmentation performance using the pixel accuracy (Pix. acc.) using equation (4) but the difference is that accuracy is additionally measured for each pixel:

$$Pix.acc. = \frac{1}{NP} \sum_{k=1}^{P} \sum_{c=1}^{N} \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \tag{7}$$

where P is the number of pixels in the segmentation mask.

B. IMAGE CLASSIFICATION EVALUATION RESULTS

We evaluated the proposed method on CIFAR10 and Tiny ImageNet test sets. We trained the proposed models using two different training strategies. The first strategy uses Mixup [46] data augmentation (Mixing two images using the alpha channel and reflecting the mixing values in the image labels by percentage) and cutout [47] data augmentation (replacing random patch from the image with a fixed value or random gaussian noise), in addition to some standard augmentations (random horizontal flipping, random crop and resize, and random rotations). The second strategy uses Randaugment [48] augmentation (a sequential probabilistic policy of various augmentations using predefined probabilities for each transformation, the transformations include translation, rotation, scale, shear, contrast, brightness, and other transformations) and cutout [47] in addition to the previously mentioned standard augmentations. The evaluation results on CIFAR10 shows that SDDS-Net attains the best top-1 accuracy (94.28%) using the second strategy of augmentation, and SDDS-Net-DW attains the second best top-1 accuracy (91.81%). The evaluation results on Tiny ImageNet show that SDDS-Net-DW attains the best top-1 accuracy (72.25%) with the second strategy of augmentation and SDDS-Net attains the second-best top-1 accuracy (71.49%). The performance of SDDS-Net is better than the other models on CIFAR10 as it has few number of labels and SDDS-Net-DW is better than the other models on Tiny ImageNet as it has a larger number of labels (200 labels).

TABLE 3. Comparison between the proposed method with the different architectures and the SOTA methods on the CIFAR10 test set for image classification in terms of parameter count and top1 accuracy. Note that the accuracies were copied from a previous research [65].

Method	PC (M)	top1-Acc
AlexNet [15]	58.32	82.00
VGG16 [30]	134.30	93.42
ResNet50 [37]	23.60	92.04
Inceptionv1 [16]	5.98	86.91
Inceptionv3 [17]	21.82	94.25
Inception-ResNet [18]	54.35	80.10
Xception [49]	20.88	79.00
DenseNet-121 [19]	7.04	91.86
MobileNetV1 [20]	3.23	77.80
MobileNetV2 [21]	2.27	79.40
NASNet [22]	4.28	82.70
ShuffleNet [23]	4.02	77.30
SD-Net	3.28	**94.28**
SD-Net-DW	3.26	91.81

C. SEMANTIC SEGMENTATION EVALUATION RESULTS

We tested our proposed architectures on both NYU depth V2 and CITYSCAPES benchmarks for the task of semantic segmentation. Table 1 shows the mIOU, Pix. acc., the parameters count (PS) in million parameters and the speed in frames per second (fps) obtained from each architecture. SDDS-Net attained the best Pix. acc. (77.5%) at a speed of 25.0 fps on NYU depth V2, and SDDS-Net-DW is ranked as the second-best accuracy however, MP-CNN and strided CNN attain acceptable values of mIOU and Pix. acc and they are less complex (as shown from the parameter count) than SDDS-Net and SDDS-Net-DW, while SDDS-Net-DW attained the best mIOU (82.35%) at 11.63 fps speed on CITYSCAPES and SDDS-Net attains an mIOU of 80.12% at 12.5 fps. Those results show that SDDS-Net is better at learning a lower number of segmentation classes (14 classes of NYU) than SDDS-Net-DW which could efficiently learn the 19 classes of the CITYSCAPES benchmark.

TABLE 4. Comparison between the proposed method with the different architectures and the SOTA methods on Tiny-ImageNet validation set for image classification in terms of parameter count and top1 accuracy. Note that the accuracies were copied from previous research [66].

Method	PC (M)	top1-Acc
ResNet-56 [37]	0.91	56.51
ResNet-18 [37]	11.60	53.32
EfficientNet B0 [26]	4.00	55.48
ResNet-110 [37]	1.70	62.56
DenseNet-121 [19]	7.05	60.00
Wide-ResNet [27]	11.00	65.99
ResNext [24]	25.00	68.23
SD-Net	3.28	71.49
SD-Net-DW	3.26	**72.25**

TABLE 5. Comparison between the proposed method with the different architectures and SOTA methods on NYU depth V2 semantic segmentation test benchmark.

Method	Pix. acc. %
Handa et al. [52]	52.5
Hermans et al. [53]	54.3
McCormac et al. [54]	59.2
Dai et al. [55]	60.7
Dai et al. [56]	71.2
Hu et al. [57]	73.5
Wang et al. [58]	78.3
SDDS-Net	**78.5**
SDDS-Net-DW	75.3

D. COMPARISON WITH SOTA METHODS ON IMAGE CLASSIFICATION

We compare the proposed classification models with the space-to-depth downsampling (SD-Net and SD-Net-DW) with the SOTA methods on CIFAR10 and Tiny ImageNet classification. Table 3 shows a comparison between the proposed models (SD-Net and SD-Net-DW) and the SOTA methods on CIFAR10 classification reporting the parameter count of each model and Top-1 accuracy. SD-Net outperforms the SOTA methods with a top-1 accuracy of 94.28% which is slightly higher than InceptionV3 top-1 accuracy (94.25%) however, it has much fewer parameters (3.28 Million parameters) than InceptionV3 (21.82 Million parameters). SD-Net-DW also attains a relatively high top-1 accuracy (91.81) outperforming most of the SOTA methods except for VGG16, ResNet50, and InceptionV3 which proves the outstanding performance of the proposed architectures. Table 4 shows a comparison between the proposed models (SD-Net and SD-Net-DW) and the SOTA methods on Tiny ImageNet classification. On this dataset, SD-Net-DW outperforms the SOTA methods with a top-1 accuracy of 72.25%. This differs from the results obtained on CIFAR10 which showed that SD-Net overpassed SD-Net-DW in the accuracy. Those results prove that each architecture has an advantage in a specific task such as learning more number of classes with a high accuracy in the case of SD-Net-DW in the Tiny ImageNet classification task against SD-Net which could learn fewer number of classes (in the case of CIFAR10 classification) with much better accuracy than SD-Net-DW.

TABLE 6. Comparison between the proposed method with the different architectures and the SOTA methods on CITYSCAPES semantic segmentation validation benchmark.

Method	mIOU %	fps
Template-Based NAS-arch1 [59]	69.5	10.0
SqueezeNAS [60]	75.2	10.2
DeepLabv3 [35]	78.5	-
PSPNet [29]	79.7	-
DeepLabv3+ [36]	79.6	1.2
HRNetV2 [61]	79.7	-
Trans4Trans [63]	81.5	4.3
CMX-B2 [64]	81.6	-
EANet [62]	81.7	-
SETR-PUP [39]	82.15	0.5
SDDS-Net	80.12	12.5
SDDS-Net-DW	**82.35**	11.6

E. COMPARISON WITH SOTA METHODS ON SEMANTIC SEGMENTATION

We compared the proposed method with the SOTA methods on semantic segmentation. SDDS-Net could outperform the SOTA methods on NYU depth V2 semantic segmentation task in terms of Pix. acc. as shown in Table 5 while our proposed architectures are much simpler than those of the SOTA methods however, SDDS-Net-DW outperforms most of the SOTA methods (almost all the SOTA methods except for the method proposed by Wang et al. [58]). SDDS-Net-DW also outperforms the SOTA methods on CITYSCAPES semantic segmentation task even the attention-based methods such as EANet [62], and HRNetV2 [61] and the transformer-based method such as Trans4Trans [63] and SETR-PUP [39] (transformers are recently one of the most efficient architectures in deep-learning) as shown in table 6 with an acceptable speed of processing (∼12 fps). Both SDDS-Net and SDDS-Net-DW attain higher speeds (12.5 fps for SDDS-Net and 11.63 fps for SDDS-Net-DW) than all other methods in the comparison.

VI. CONCLUSION

The proposed architectures can efficiently learn the image classification as a result of using the powerful SD module for the lossless image down-sampling instead of the traditional pooling and strided convolution methods. It also could learn the dense prediction task of semantic segmentation as a result of using the SD module in the encoder stage and the DS module for up-sampling in the decoder stage. The evaluation results on CIFAR10 and Tiny ImageNet classification tasks show the superior performance of the proposed SD module for downsampling (SD-Net attains 94.28% and 71.49% on CIFAR10 and Tiny ImageNet, respectively, and SD-Net-DW attains 91.81% and 72.25% on the same benchmarks) in addition to the efficient design of the architectures. The proposed SD-Net and SD-Net-DW outperform the SOTA methods in Image classification while the model consists of relatively a few number of parameters. Additionally, the proposed SDDS-Net and SDDS-Net-DW could perform the task of segmentation with high speed which is convenient for real-time applications. The strength of the proposed method was proved by the evaluation results on NYU depthV2 and CITYSCAPES semantic segmentation results (SDDS-Net attains Pix. acc of 78.55% for NYU depthV2, and 85.01% for CITYSCAPES, and SDDS-Net-DW attains a Pix. acc. value of 75.26 for NYU depthV2 and 87.9 for CITYSCAPES) as the proposed architectures based on SD and DS modules outperform the SOTA methods on both NYU depthV2 and CITYSCAPES benchmarks. In this work, we focused on the optimization of the encoder architecture of the encoder-decoder model for semantic segmentation however, in our future work, we aim to design an architecture that depends on multiple stages of the depth-to-space module in the decoder instead of the single stage we applied in the proposed architectures in this paper. We think this approach can improve the performance of the semantic segmentation models.

REFERENCES

[1] R. Ayachi, M. Afif, Y. Said, and M. Atri, "Strided convolution instead of max pooling for memory efficiency of convolutional neural networks," in *Proc. 8th Int. Conf. Sci. Electron., Technol. Inf. Telecommun. (SETIT)*, vol. 1, 2020, pp. 234–243.

[2] C. Yang, Y. Wang, X. Wang, and L. Geng, "A stride-based convolution decomposition method to stretch CNN acceleration algorithms for efficient and flexible hardware implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 3007–3020, Sep. 2020.

[3] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[4] J. V. Hurtado and A. Valada, "Semantic scene segmentation for robotics," in *Deep Learning for Robot Perception and Cognition*. New York, NY, USA: Academic Press, 2022, pp. 279–311.

[5] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic segmentation of 3D point clouds," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 537–547.

[6] M. Z. Khan, M. K. Gajendran, Y. Lee, and M. A. Khan, "Deep neural architectures for medical image semantic segmentation: Review," *IEEE Access*, vol. 9, pp. 83002–83024, 2021, doi: 10.1109/ACCESS.2021.3086530.

[7] T.-Y. Ko and S.-H. Lee, "Novel method of semantic segmentation applicable to augmented reality," *Sensors*, vol. 20, no. 6, p. 1737, Mar. 2020.

[8] A. Aliouat, N. Kouadria, S. Harize, and M. Maimour, "An efficient low complexity region-of-interest detection for video coding in wireless visual surveillance," *IEEE Access*, vol. 11, pp. 26793–26806, 2023, doi: 10.1109/ACCESS.2023.3248067.

[9] A. Aliouat, N. Kouadria, M. Maimour, S. Harize, and N. Doghmane, "Region-of-interest based video coding strategy for rate/energy-constrained smart surveillance systems using WMSNs," *Ad Hoc Netw.*, vol. 140, Mar. 2023, Art. no. 103076, doi: 10.1016/J.ADHOC.2022.103076.

[10] Q. Sellat, S. K. Bisoy, and R. Priyadarshini, "Semantic segmentation for self-driving cars using deep learning: A survey," in *Cognitive Big Data Intelligence With a Metaheuristic Approach*. New York, NY, USA: Academic Press, 2022, pp. 211–238.

[11] L. Wang, Y. Guo, L. Liu, Z. Lin, X. Deng, and W. An, "Deep video super-resolution using HR optical flow estimation," *IEEE Trans. Image Process.*, vol. 29, pp. 4323–4336, 2020.

[12] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.

[13] Z. Zhang, Z. Cui, C. Xu, Z. Jie, X. Li, and J. Yang, "Joint task-recursive learning for RGB-D scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2608–2623, Oct. 2020.

[14] X. Lin, D. Sánchez-Escobedo, J. R. Casas, and M. Pardàs, "Depth estimation and semantic segmentation from a single RGB image using a hybrid convolutional neural network," *Sensors*, vol. 19, no. 8, p. 1795, Apr. 2019.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1–9.

[16] 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.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, Feb. 2017, pp. 1–7.

[19] 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.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[22] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[23] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[24] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

[25] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11976–11986.

[26] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[27] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.

[28] 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.

[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–6.

[31] 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.

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.

[33] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–9.

[34] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

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

[36] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 833–851.

[37] 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.

[38] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[39] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6877–6886.

[40] S. Wu, T. Wu, F. Lin, S. Tian, and G. Guo, "Fully transformer networks for semantic image segmentation," 2021, *arXiv:2106.04108*.

[41] H. Ibrahem, A. Salem, and H.-S. Kang, "DTS-net: Depth-to-space networks for fast and accurate semantic object segmentation," *Sensors*, vol. 22, no. 1, p. 337, Jan. 2022.

[42] H. Ibrahem, A. Salem, and H.-S. Kang, "DTS-depth: Real-time single-image depth estimation using depth-to-space image construction," *Sensors*, vol. 22, no. 5, p. 1914, Mar. 2022.

[43] H. Ibrahem, A. Salem, and H.-S. Kang, "LEOD-net: Learning line-encoded bounding boxes for real-time object detection," *Sensors*, vol. 22, no. 10, p. 3699, May 2022, doi: 10.3390/S22103699.

[44] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, USA, Tech. Rep. 0, 2009.

[45] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, pp. 1–6, 2015.

[46] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[47] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.

[48] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 702–703.

[49] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[50] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 746–760.

[51] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[52] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding RealWorld indoor scenes with synthetic data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4077–4085.

[53] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from RGB-D images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 2631–2638.

[54] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semantic Fusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4628–4635.

[55] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2432–2443.

[56] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 458–474.

[57] W. Hu, H. Zhao, L. Jiang, J. Jia, and T.-T. Wong, "Bidirectional projection network for cross dimension scene understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14373–14382.

[58] Z. Wang, Y. Rao, X. Yu, J. Zhou, and J. Lu, "SemAffiNet: Semantic-affine transformation for point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11819–11829.

[59] V. Nekrasov, C. Shen, and I. Reid, "Template-based automatic search of compact semantic segmentation architectures," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1969–1978.

[60] A. Shaw, D. Hunter, F. Landola, and S. Sidhu, "SqueezeNAS: Fast neural architecture search for faster semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2014–2024.

[61] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[62] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, "Beyond self-attention: External attention using two linear layers for visual tasks," 2021, *arXiv:2105.02358*.

[63] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen, "Trans4Trans: Efficient transformer for transparent object and semantic scene segmentation in real-world navigation assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19173–19186, Oct. 2022.

[64] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, "CMX: Cross-modal fusion for RGB-X semantic segmentation with transformers," 2022, *arXiv:2203.04838*.

[65] N. Darapaneni, B. Krishnamurthy, and A. R. Paduri, "Convolution neural networks: A comparative study for image classification," in *Proc. IEEE 15th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Nov. 2020, pp. 327–332.

[66] K. Patel and G. Wang, "A discriminative channel diversification network for image classification," *Pattern Recognit. Lett.*, vol. 153, pp. 176–182, Jan. 2022.

HATEM IBRAHEM received the B.Eng. degree in electrical engineering (electronics and communication) from Assiut University, Egypt, in 2013, and the Ph.D. degree in information and communication engineering from the School of Electrical and Computer Engineering, Chungbuk National University, Cheongju, South Korea, in 2023. He is currently a Postdoctoral Fellow with the Department of Information and Communication Engineering, Chungbuk National University. His research interests include multimedia, image processing, machine learning, deep learning, and computer vision.

AHMED SALEM received the B.Eng. degree in electrical engineering (electronics and communication) from Assiut University, Egypt, in 2012, the M.Eng. degree in electronics and communication engineering from the Egypt–Japan University of Science and Technology, Egypt, in 2016, and the Ph.D. degree in information and communication engineering from the School of Electrical and Computer Engineering, Chungbuk National University, Cheongju, South Korea, in 2022. He is currently a Postdoctoral Fellow with the Department of Information and Communication Engineering, Chungbuk National University. His research interests include multimedia, image processing, machine learning, deep learning, and computer vision.

HYUN-SOO KANG (Member, IEEE) received the B.S. degree in electronic engineering from Kyoungpook National University, Republic of Korea, in 1991, and the M.S. and Ph.D. degrees in electrical and electronics engineering from KAIST, in 1994 and 1999, respectively. From 1999 to 2005, he was with Hynix Semiconductor Company Ltd., the Electronics and Telecommunications Research Institute (ETRI), and Chung-Ang University. In March 2005, he joined the College of Electrical and Computer Engineering, Chungbuk National University, Chungbuk, Republic of Korea. His research interests include image compression and image processing.