



A new CNN-based semantic object segmentation for autonomous vehicles in urban traffic scenes

Gürkan Doğan¹ · Burhan Ergen²

Received: 1 January 2023 / Revised: 25 September 2023 / Accepted: 29 November 2023 / Published online: 23 February 2024
© The Author(s) 2024

Abstract

Semantic segmentation is the most important stage of making sense of the visual traffic scene for autonomous driving. In recent years, convolutional neural networks (CNN)-based methods for semantic segmentation of urban traffic scenes are among the trending studies. However, the methods developed in the studies carried out so far are insufficient in terms of accuracy performance criteria. In this study, a new CNN-based semantic segmentation method with higher accuracy performance is proposed. A new module, the Attentional Atrous Feature Pooling (AAFP) Module, has been developed for the proposed method. This module is located between the encoder and decoder in the general network structure and aims to obtain multi-scale information and add attentional features to large and small objects. As a result of experimental tests with the CamVid data set, an accuracy value of approximately 2% higher was achieved with a mIoU value of 70.59% compared to other state-of-art methods. Therefore, the proposed method can semantically segment objects in the urban traffic scene better than other methods.

Keywords Autonomous vehicles · Street scenes · Traffic scenes · Pixel-wise semantic segmentation · Deep learning · Computer vision

1 Introduction

The ability of vehicles to perceive the surrounding environment with high performance is a very critical task for the transition to fully autonomous driving systems. Especially in crowded urban traffic scenes, there is a very diverse ecosystem such as pedestrians, vehicles, road boundaries, trees, buildings, etc. Accident-free travel of vehicles in a fully autonomous manner within this complex ecosystem is very important in terms of driving and vehicle safety, pedestrian safety, and continuity of traffic flow. In order for autonomous vehicles to drive accident-free, that is, safely, in a real traffic environment, software subsystems must be designed perfectly. An autonomous vehicle has a minimum

of 5 software subsystems [1, 2]; localization, perception, planning, vehicle control and system management. Localization is responsible for estimating vehicle location. Perception derives a model of the driving environment from multi-sensor fusion-based information. Based on localization and perception information, the planning subsystem determines the vehicle's maneuvers for safe vehicle navigation. The vehicle control subsystem follows the command requested from the planning subsystem by steering, accelerating and braking the autonomous vehicle. Finally, system management oversees the overall autonomous driving system. Perception, one of these software subsystems, carries out the tasks of identifying, detecting and segmenting objects in the dynamic environment around the vehicle using sensor components such as cameras, radars and laser scanners. The information obtained by this subsystem is very important in terms of planning the behavior of the autonomous vehicle, providing navigation and control of the vehicle. However, in a dynamic and complex traffic environment, it is a big problem for the perception subsystem to identify, detect and segment objects with high accuracy. This problem can only be solved by autonomous vehicles detecting objects with high accuracy,

✉ Gürkan Doğan
gurkandogan@munzur.edu.tr
Burhan Ergen
bergen@firat.edu.tr

¹ Department of Computer Engineering, Faculty of Engineering, Munzur University, Tunceli, Turkey

² Department of Computer Engineering, Faculty of Engineering, Firat University, Elazig, Turkey

because incorrect object detection is the factor that increases the risk of accidents.

Nowadays, semantic segmentation methods based on deep learning are used for the detection of objects in urban traffic scenes [3–5]. Semantic segmentation aims to perform pixel-wise image classification detailed enough to assign predefined labels corresponding to each pixel in an input image. At the same time, semantic segmentation is one of the main challenging tasks in the field of computer vision, as it requires pixel-by-pixel accuracy and implements multi-scale contextual reasoning. However, due to the rich hierarchical features and detailed object information provided by semantic segmentation, it is an important method for the knowledge of the urban traffic environment and for making sense of the relationship between objects. Also, semantic segmentation can predict the category, position and shape of each element [4]. In particular, it is widely used in applications that require estimating the precise boundaries of an object [6–9].

Traditional methods for semantic segmentation often rely on handcraft methods such as [10–12]. However, the performance of such traditional methods is not satisfactory compared to modern methods. Modern methods used today are based on deep learning and even when these methods are applied for different data sets, their performance is very high. Most modern methods are based on fully convolutional networks (FCN) [13]. This method uses the full convolutional layer and directly predicts probability maps for each class. Besides, a deep learning-based semantic segmentation algorithm usually consists of an encoder and a decoder. The encoder is the network that extracts the low-resolution features, while the decoder is the network that generates the multi-dimensional features from the features extracted by the encoder [14, 15]. SegNet [15] proposed by Badrinarayanan et al. [16] only stores the maximum pooling indices of attribute maps and uses them in the decoder network to reduce memory and time complexity at the expense of accuracy. In the U-Net method proposed by Ronneberger et al., it adds the feature maps obtained from each layer of the encoder to the decoder and thus improves the performance. The Atrous Spatial Pyramid Pooling module was developed in the DeepLabv3+ [17] method proposed by Chen et al. [18]. Thanks to this module, object boundaries are better determined by coding multi-scale contextual information and using spatial information gradually. Besides, semantic segmentation algorithms such as DeepLabv3+, Global Convolutional Network (GCN), dense upsampling convolution (DUC) [19], and PSPNet [20], etc., use dilated convolution or a large convolution kernel to further preserve spatial position information. The RefineNet [21] algorithm proposed by Lin et al. uses a multi-path refinement network that leverages all available information along with downsampling to generate high-resolution predictions. The DCNN-based method [3] proposed by Dong et al. provides a good balance between

accuracy and speed, while it is claimed by the authors to be real time and high performance. The proposed method uses atrous convolution, a convolutional attention module, and a lightweight network to extract features efficiently and obtain dense feature maps. It also uses rich semantic information to overcome the multi-scale problem of semantic segmentation. A new method has been proposed in which the Global Encoding Module (GEModule) and the Dilated Decoder Module (DDModule) developed by Fan et al. [4] are used together. GEModule is used to select distinctive feature maps, while DDModule combines dilated convolution and dense connections to improve prediction results. In the JPANet [5] method proposed by Hu et al., a joint feature pyramid module was developed to capture the shallow network multi-level local features, improving pixel classification performance, and a spatial detail extraction module to preserve geometric information lost during the downsampling phase. In addition, they developed the bilateral feature fusion module to fuse high-level semantic features and low-level spatial features with information complementarity by establishing the dependencies of feature information in channel dimension and position dimension. However, one of the main difficulties for these methods proposed in the literature is that, due to the similarity between objects and the existence of multi-scale objects, there were great errors in the prediction results between objects. Therefore, although many methods are proposed in different architectures, the accuracy performance is insufficient for pixel-based detection of objects.

In this article, Attentional Feature Pooling (AFP) Module and Attentional Atrous Feature Pooling (AAFP) Module are proposed specifically for the general architecture to overcome this difficulty. The AFP Module basically consists of three stages. In the first stage, the feature map taken as input is passed through the maximum and average pooling blocks three times in order to create different sizes and obtain important features. Then, the features extracted from the maximum and average pooling are combined. In the second step, these combined features are multiplied with one feature obtained from them through the sigmoid block. In this way, attentional features are obtained. In the third stage, the attentional features obtained in 3 different dimensions are applied upsampling to obtain the input size. The AAFP Module consists of two stages. In the first stage, convolution blocks with different dilated ratios (dr; 1, 6, 12, 18) are applied. Dilated convolution is used to obtain multi-scale information. In the second step, the AFP Module is applied and the three outputs of the AFP Module and the outputs of the convolution blocks are combined and passed through the convolution block once again. In the general architecture of the proposed method, the ResNet50 CNN model, which was pre-trained with ImageNet, was used as the encoder. The decoder has learned from the features obtained from the

middle and last layers of ResNet50. In the proposed architecture, the AAFP Module is used to learn the detailed texture information from the features obtained from the last layer of ResNet50 and transfer it to the decoder. The decoder consists of upsampling, dropout, convolution, concatenate, and convolution block structures/operators.

In summary, the contributions of this study are as follows;

- The Attentional Feature Pooling (AFP) Module is proposed to obtain information of smaller objects and to draw attention to the attributes of these objects.
- The Attentional Atrous Feature Pooling (AAFP) Module has been developed by adding dilated convolution blocks to the AFP Module to obtain multi-scale information about objects.
- The overall architecture was designed based on the AAFP Module for semantic image segmentation in urban traffic scenes.
- The proposed architecture also contributes to reducing the risk of accidents of autonomous vehicles as it provides higher segmentation accuracy than state-of-the-art methods.
- A series of experimental tests were conducted to demonstrate the effectiveness of the proposed architecture. And, the comparison of the proposed method/architecture with respect to other methods/architectures was made with the CamVid dataset.

The remainder of the paper is organized as follows. In chapter 2 contains information about the dataset, encoder of the proposed network model, learning rate scheduling, and dice loss. In the 3rd chapter, the proposed method and in the 4th chapter, the experiments and analysis are given. Finally, the general conclusions are presented in the 5th chapter.

2 Materials and methods

In this section, the CamVid dataset and learning rate scheduling used in the training and testing of the proposed model and other state-of-the-art models are introduced. ResNet50, which is used to extract features in the encoder network of the proposed model, is briefly introduced. Finally, Dice Loss is introduced to calculate the losses of the proposed model and other state-of-the-art models during training.

2.1 CamVid dataset

The CamVid Dataset [22] is a publicly available dataset containing street scenes captured from the perspective of a car driver. The original images of the dataset contain 32 semantic categories and 701 images with a resolution of 960×720 . Of the 701 images, 367 are reserved for training, 101 for

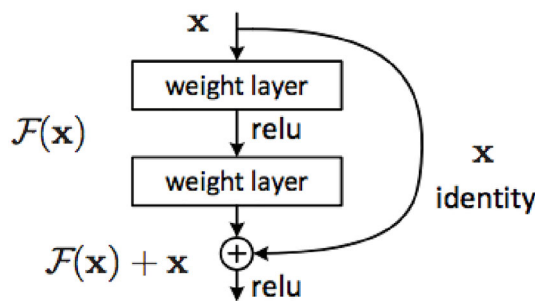


Fig. 1 Skip/residual connection

validation, and 233 for testing. The resolution of the images in the CamVid dataset was set to 256×256 to perform the performance measurements of the proposed method. Twelve semantic categories were used, namely sky, building, pole, road, pavement, tree, sign symbol, fence, car, pedestrian, bicyclist, and unlabeled.

2.2 ResNet50 encoder

In order to ensure that the performance of the proposed method is strong, transfer learning Residual Network (ResNet50) [23], which is a good classification solution, is used. ResNet50 tries to solve the vanishing gradient problem, which is one of the difficulties in the training process of deep CNN models. During backpropagation, the gradient value is significantly reduced and therefore there is almost no change in weights. ResNet50 uses skip/residual connection to overcome this. With this connection, the original output is added to the output of the convolutional block. That is, it is a link that directly bypasses some layers of the network model.

The skip connection is schematized in Fig. 1. As can be seen in this figure, with residual links, layers are allowed to fit a match and the desired fundamental match is expressed as $H(x)$. The stacked nonlinear layers are also allowed to fit another match, expressed as $F(x) := H(x) - x$. Thus, the original mapping is recast as $H(x) := F(x) + x$. In this way, shortcut identity mapping ensures that no additional parameters are added to the model and the computation time is kept under control.

The residual networks used in this study have 50 layers and this network model is called ResNet50. The general architecture of ResNet50 is presented in Fig. 2. As can be seen in this figure, a 7×7 size and 64-kernel convolution operation are performed in the first layer. Then comes the 3×3 maximum pooling layer. These first two layers are used $\times 1$ times, while the third layer block uses 1×1 size and 64-kernel convolution, 3×3 size and 64-kernel convolution, and 1×1 size and 256-kernel convolution layers $\times 3$ times, respectively. Therefore, there are 9 layers in total

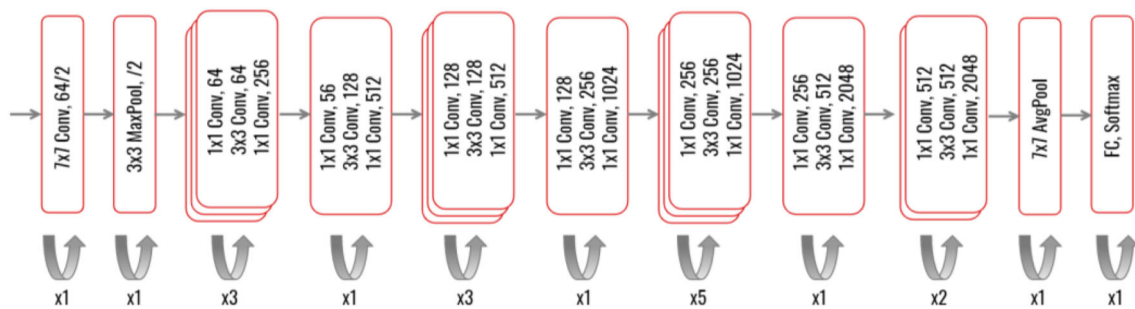


Fig. 2 ResNet50 architecture

in this layer block. Other layer blocks have similar use. It also has residual connections given in Fig. 1, although not expressed in the network architecture. On the other hand, the last layer of the ResNet50 model used in the proposed method is “conv4_block6_2_relu”. While the top blockless structure of a standard ResNet50 with a $256 \times 256 \times 3$ input has a total parameter number of 23.58 Million, the ResNet50 used in the proposed method has 8.32 Million parameters. The reason why the number of parameters is less in the proposed method is the selection of “conv4_block6_2_relu” as the last layer of ResNet50.

2.3 Learning rate scheduling

In this study, Adam [24] was used as the optimizer. Adam optimizer is a stochastic gradient descent algorithm based on the estimation of first-order and second-order moments. The reason for choosing this optimizer is that it is computationally efficient and requires less memory consumption and is more suitable for problems with a large number of parameters. On the other hand, the “ReduceLROnPlateau” function of the Keras deep learning API [25] is used for learning rate scheduling together with the Adam optimizer. This function reduces the learning rate by a multiple of the factor parameter determined when the learning of the network model stagnates. The stagnation of learning is decided by monitoring the validation loss parameter as much as the number of patience. In this way, learning of a network model is more efficient.

The parameters used in the proposed method are given in Table 1. Accordingly, validation loss, patience 20, factor 0.3, and minimum learning rate $1e-5$ for learning rate scheduling are set.

2.4 Dice loss

Dice loss is derived from a statistically based Sørensen–Dice coefficient developed in the 1940s to measure similarity between two objects [26]. It was developed in 2016 by Milletari et al. [27] for three-dimensional medical image segmentation. Dice coefficient (DC) is a function that determines the similarity measure in binary classification tasks. Since $DC \in [0, 1]$, the value of the dice coefficient varies in this range, including 0 (zero) and 1 (one). The dice coefficient for binary classification in image segmentation tasks is calculated by the following equation;

$$DC = \frac{2 \sum_i^N p_i g_i + \varepsilon}{\sum_i^N p_i^2 + \sum_i^N g_i^2 + \varepsilon} \quad (1)$$

In Eq. (1), p_i , the predicted probability of the i -th pixel/voxel, g_i , the ground truth of the i -th pixel/voxel, N , the number of pixels/voxels and i denotes index of pixel/voxel. ε is the minimum value that prevents the zero denominators from occurring. $\sum_i^N p_i g_i$ is the probability of intersection of predicted and ground truth pixels/voxels. $\sum_i^N p_i^2$ is the sum of squares of the probabilities of the predicted pixels/voxels. $\sum_i^N g_i^2$ is the sum of squares of the ground truth pixels/voxels. In a binary image segmentation task, it can be expressed as $g_i \in \{0, 1\}$. The dice loss equation is expressed as follows.

$$DC_{loss} = 1 - DC \quad (2)$$

The reason for using the Dice Loss function in this study is that it is easy in terms of computation and has extensive and effective use in applications such as intelligent transportation systems and medical image processing.

Table 1 Details about parameters of the proposed method

Parameters	Epoch	Batch	Optimizer	Learning Rate Sche	Monitor	Patience	Factor	Min LR
Values	300	8	Adam	ReduceLROnPlateau	val. loss	20	0.3	$1e-5$

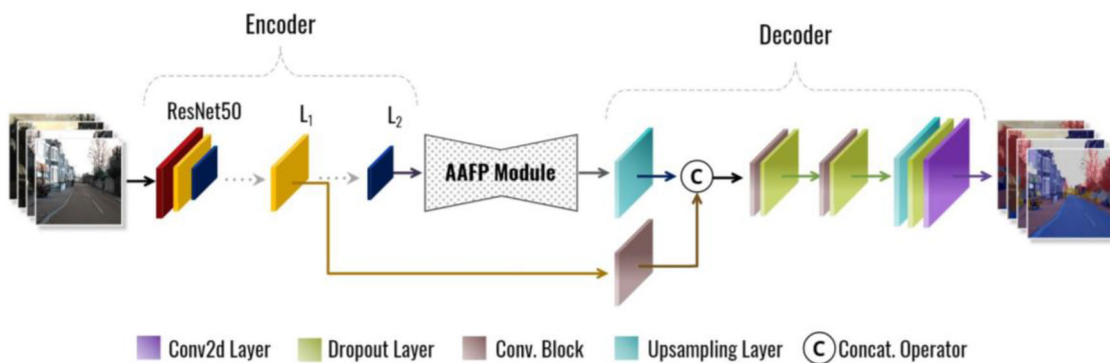


Fig. 3 Architecture of the proposed method

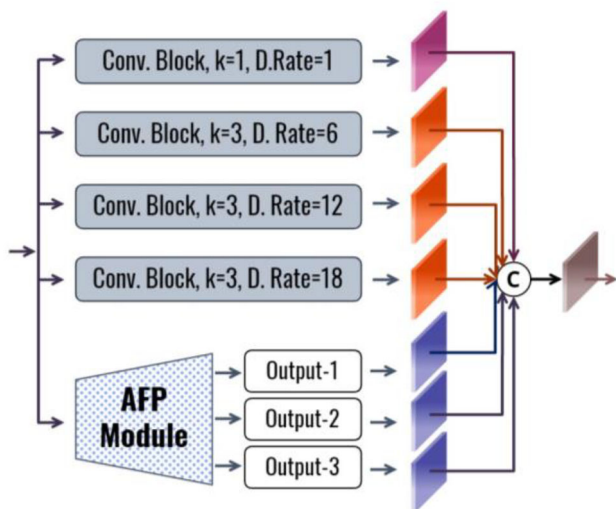


Fig. 4 Attentional Atrous Feature Pooling Module

3 The proposed method

In the proposed method, the detection of objects for autonomous driving in urban traffic scenes is considered as a pixel-wise classification task and the architecture of the proposed method is given in Fig. 3. As seen in this figure, the pre-trained ResNet50 CNN model, whose success in classification is known, is used in the encoder block of the network architecture. The convolution block, upsampling layer, dropout layer, and convolution layers are used in the decoder block. A newly developed Attentional Atrous Feature Pooling (AAFP) Module is used between the encoder and decoder blocks. The AAFP module is presented in Fig. 4.

The AAFP Module is used to obtain multi-scale information and to add attentional features to objects in large and small feature maps. The AAFP Module consists of two different structures. The first structure is the dilated convolution block. In this structure, convolution blocks are applied at different dilated ratios such as 1, 6, 12, and 18. If the dilated

rate is 1, this structure is also called a standard convolution block. Dilated convolution block is used to obtain multi-scale information. The second structure used in the AAFP Module is the Attention Feature Pooling (AFP) Module in Fig. 5. This structure is used to add attentional features to objects in large and small-size feature maps. The AFP Module basically consists of three stages. In the first stage, the feature map taken as input is passed through the maximum and average pooling blocks given Fig. 6, three times to create different dimensions and obtain important features. The purpose of creating feature maps of different dimensions is to reach even more detailed information of small and large size objects. Then, the features extracted from the maximum and average pooling are combined. In the second stage, these combined features are multiplied with one feature obtained from them through the sigmoid block given in Fig. 6. In this way, the objects in the feature maps are clarified and attentional features are obtained. In the third stage, the attentional features obtained in 3 different dimensions are applied upsampling to obtain the input size. Thus, it is ensured that the input size of the AFP Module is the same as the output size. The block structures of the AFP Module are presented in Fig. 5. In addition, the outputs of the dilated convolution blocks used in the AAFP Module and the AFP Module are combined and passed through the convolution block once again. At this point, the reason for using the convolution block again is to reduce the number of combined features and predict new features.

Additionally, in the proposed method, L_1 (conv2_block3_2_relu), one of the middle layers of ResNet50 in the encoder block, is transferred to the decoder block by passing through the convolution block with a skip connection, while L_2 (conv4_block6_2_relu), the last layer of ResNet50, is transferred to the decoder block after passing through the AAFP Module. On the other hand, the number of outputs and parameters obtained according to the operators, hyperparameters and inputs used in the block structures of the proposed method are given in Table 2.

Fig. 5 Attentional Feature Pooling Module

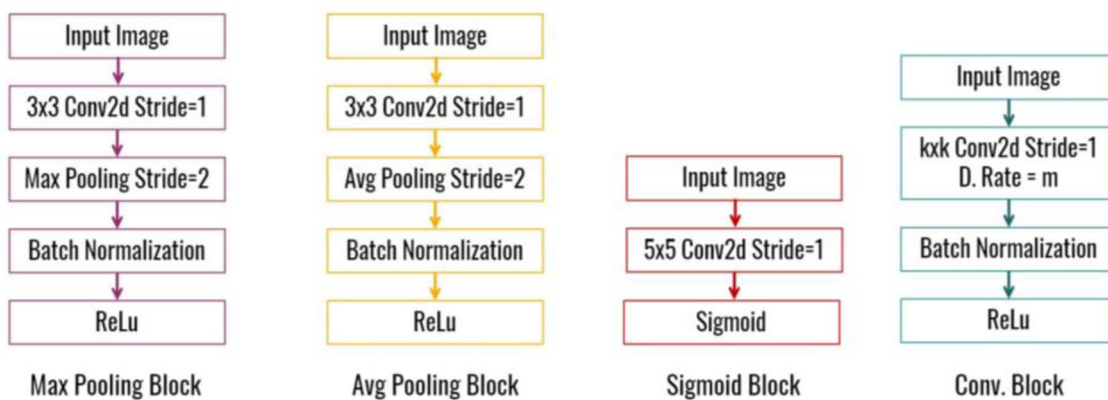
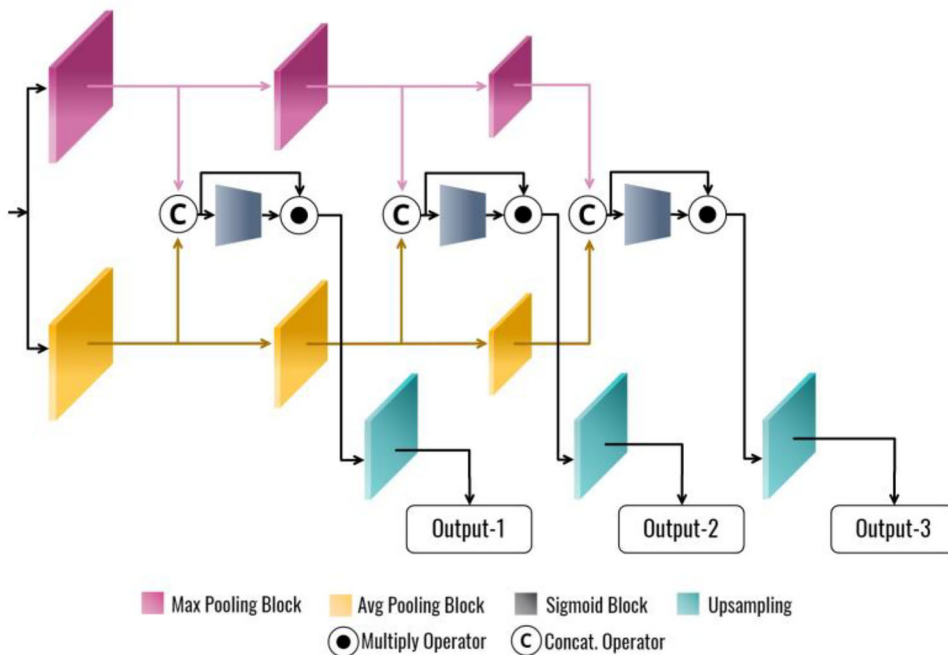


Fig. 6 Block structures used in the proposed method

4 Experiments and analysis

In this study, training and testing of the proposed method for the semantic segmentation of urban traffic scenes were performed with the CamVid dataset. No data augmentation method was applied for this data set. The reason for this is to better observe the effect of the proposed method on performance. While the batch size is set to 8 for the training processes of the model, the epoch is set to 300. Adam optimizer and ReduceLRonPlateau functions were used together to optimally realize the learning rate of the model and reduce the number of computational complexities. The software and hardware technologies used in the training and testing processes of the model are as follows; Intel(R) Xeon(R) CPU @ 2.30 GHz, 24 GB RAM, Tesla P100-PCIE-16 GB, Python version 3.7.13, Tensorflow version 2.8.2, Keras version 2.8.0.

On the other hand, mean intersection over union (mIoU), the number of parameters (NP), and frame per second (FPS) metrics were used for the performance tests of the proposed method.

mIoU is a metric used to measure the percentage of overlap of pixels between the predicted mask and the actual mask. If the value of the mIoU metric is close to 1 (i.e. 100%), the mask predicted by the network model is very similar to the actual mask. If the value of this metric is close to 0 (i.e. 0%), there is little similarity between the mask predicted by the network model and the actual mask. In Fig. 7, the IoU formula and its schematic representation are expressed [28]. Accordingly, TP denotes true-positive, FN false-negative, and FP false-positive.

Table 2 The block structures in the Attentional Atrous Feature Pooling Module (AAFP), k denotes kernel filter size, sp pooling size, c channel number, s stride size, dr dilation rate, and n denotes the number of times the operator

Blocks	Inputs	Operators	Outputs	n	Parameters
Conv. Block	$h \times w \times c_1$	$1 \times 1, Conv2d, s = 1$	$h \times w \times c_2$	1	$c_2 \times (c_1 \times 1 \times 1)$
	$h \times w \times c_2$	<i>BatchNorm, Relu</i>	$h \times w \times c_2$		$4 \times c_2$
Dilated Conv. Block	$h \times w \times c_1$	$3 \times 3, Conv2d, s = 1, dr = [6, 12, 18]$	$h \times w \times c_2$	3	$c_2 \times (c_1 \times 3 \times 3)$
	$h \times w \times c_2$	<i>BatchNorm, Relu</i>	$h \times w \times c_2$		$4 \times c_2$
Avg. Pooling Block	$h \times w \times c_1$	$3 \times 3, Conv2d, s = 1$	$h \times w \times c_2$	3	$c_2 \times (c_1 \times 3 \times 3 + 1)$
	$h \times w \times c_2$	<i>AvgPooling2d, $s_p = 2$</i>	$(h/s_p) \times (w/s_p) \times c_2$		0
	$(h/s_p) \times (w/s_p) \times c_2$	<i>BatchNorm, Relu</i>	$(h/s_p) \times (w/s_p) \times c_2$		$4 \times c_2$
Max. Pooling Block	$h \times w \times c_1$	$3 \times 3, Conv2d, s = 1$	$h \times w \times c_2$	3	$c_2 \times (c_1 \times 3 \times 3 + 1)$
	$h \times w \times c_2$	<i>MaxPooling2d, $s_p = 2$</i>	$(h/s_p) \times (w/s_p) \times c_2$		0
	$(h/s_p) \times (w/s_p) \times c_2$	<i>BatchNorm, Relu</i>	$(h/s_p) \times (w/s_p) \times c_2$		$4 \times c_2$
Sigmoid Block	$h \times w \times (2 \times c_2)$	$5 \times 5, Conv2d, s = 1, Sigmoid$	$h \times w \times 1$	3	$1 \times ((2 \times c_2) \times 5 \times 5)$
-	$(h/s_p) \times (w/s_p) \times (2 \times c_2)$	<i>Upsampling2d, s_p</i>	$h \times w \times (2 \times c_2)$	3	0
Conv. Block	$h \times w \times (10 \times c_2)$	$1 \times 1, Conv2d, s = 1$	$h \times w \times c_2$	1	$c_2 \times ((10 \times c_2) \times 1 \times 1)$
	$h \times w \times c_2$	<i>BatchNorm, Relu</i>	$h \times w \times c_2$		$4 \times c_2$

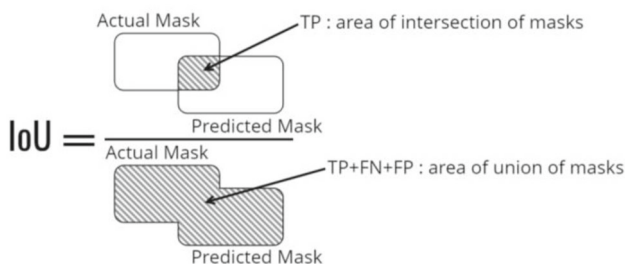


Fig. 7 Schematic representation of the iou formula

Another metric, FPS, indicates the number of frames the network model predicts per second. This is another metric used to measure device performance.

When the proposed method is trained with the CamVid dataset, accuracy, loss, dice, and Jaccard coefficient charts are generated as in Fig. 8. While the accuracy value for the validation set in the accuracy chart was approximately 90%, it reached approximately 96% for the training set. In the loss chart, the loss value for the validation set was approximately 0.1, while it reached approximately 0.04 for the training set. In the dice coefficient graph, while the dice coefficient was approximately 90% for the validation set, it reached approximately 96% for the training set. In the Jaccard coefficient chart, while the Jaccard Coefficient was approximately 82% for the validation set, it reached approximately 92% for the training set. The Jaccard Coefficient is also important to use during training as it is a metric that measures the similarity between two images.

The class-based mIoU comparison of the proposed method with other methods is given in Table 3. Unet, PSP-Net and DeepLabV3+ methods from other methods were re-implemented, and the same configuration settings and data set as the proposed method were used. In this way, only the impact of network architectures in terms of mIoU metric could be clearly observed. Furthermore, the proposed method was approximately 2% more successful in class-based mIoU than its closest competitor, DeepLabV3+. On the other hand, in order to demonstrate the effect of the AAFP module developed in the proposed method, experimental training and tests were carried out by removing the AAFP module in the proposed method. The impact of the AAFP module removed from the proposed method is given in Table 3. Accordingly, the AAFP module increased the segmentation accuracy of the proposed model by approximately 0.5%.

The comparison of the proposed method with the start of the art methods in the literature is given in Table 4. On the other hand, JPanet-M method has the best FPS value with 256, and the FSSNet method has the best parameter number with 0.2 Million. However, the proposed method has an approximately 2.3% higher mIoU score than the JPanet-M method, while it has an approximately 12% higher mIoU score than the FSSNet method. The proposed method has increased the accuracy success (mIoU) without compromising the number of parameters and FPS values too much at the expense of accuracy. However, the proposed method is advantageous as the number of parameters is less than methods such as Unet, SegNet and ICNet. Likewise, the FPS value of the proposed method is higher than the methods such as SegNet, Unet and ICNet.

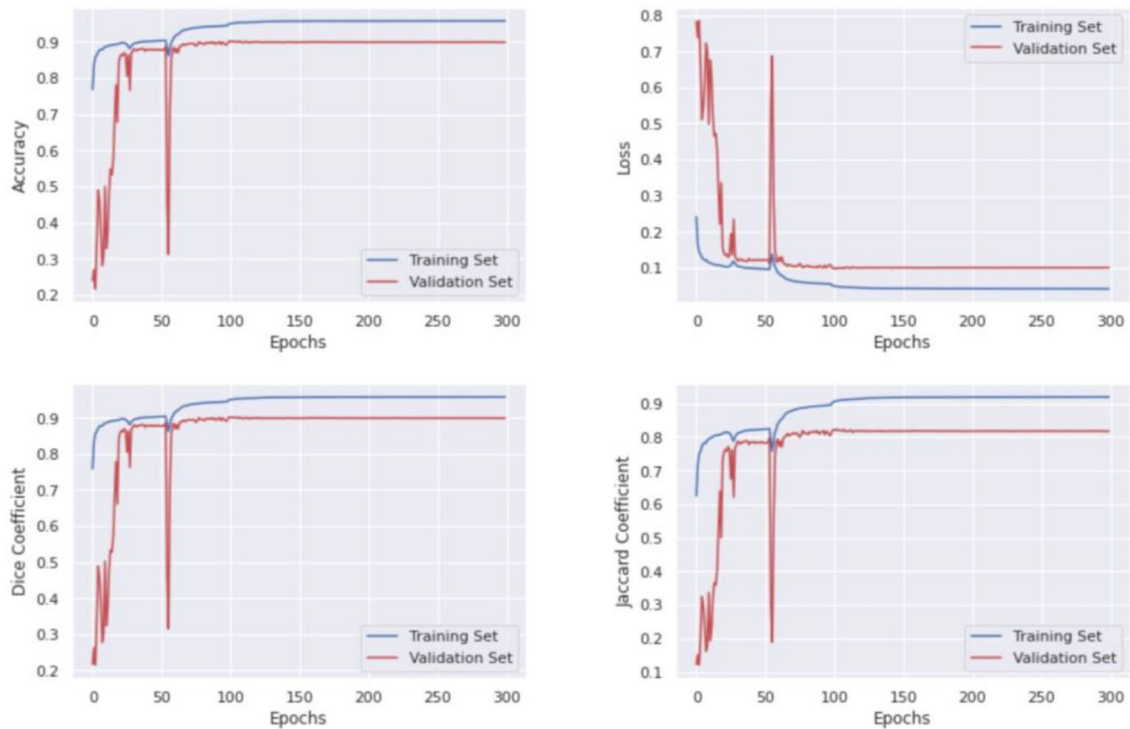


Fig. 8 Accuracy, loss, jaccard, and dice coefficient curves of the proposed method

Table 3 Class mIoU scores on the CamVid test set for the per-class

Method	Sky	Buil	Pole	Road	Pav	Tree	Sign	Fen	Car	Ped	Bic	Unl	mIoU
Unet [16]	93.76	77.73	49.41	92.37	83.48	73.33	49.49	49.41	80.22	49.68	49.90	48.10	66.41
PSPNet [20]	93.24	80.41	49.41	91.78	82.26	78.24	49.49	49.41	82.43	49.68	49.90	48.10	67.03
DeepLabV3+ [17]	93.48	79.79	49.41	93.85	84.46	79.25	53.26	49.41	83.29	49.68	49.90	60.87	68.89
Prop. Method without AAFP	93.63	81.10	49.41	93.97	85.13	80.43	65.81	49.41	82.82	49.68	49.90	59.68	70.08
Prop. Method	93.49	81.24	49.41	94.41	85.44	80.10	58.65	57.70	85.41	49.68	49.90	61.64	70.59

Figure 9 shows the qualitative visual results of the proposed method and other methods that have been re-implemented, including successful and unsuccessful predictions with the CamVid test data set. As seen in this figure, the proposed method was able to correctly assign labels for different objects in a complex urban traffic environment (multi-scale variations, illumination changes, occlusions, etc.). Examples of this situation are objects such as roads, buildings, sky, and pavements. However, sometimes incorrect segmentation estimates can be made for objects, i.e. objects can be sub-segmented or over-segmented. Examples of this situation are the car image in the first row and the bicycle objects in the second row, respectively. The main reason for the occurrence of this condition is that the appearance inside objects differs significantly. The fact that an object has different appearances is thought to be largely due

to the difference in its illumination. For such reasons, the problem of over-segmentation arises. Conversely, it causes sub-segmentation problems.

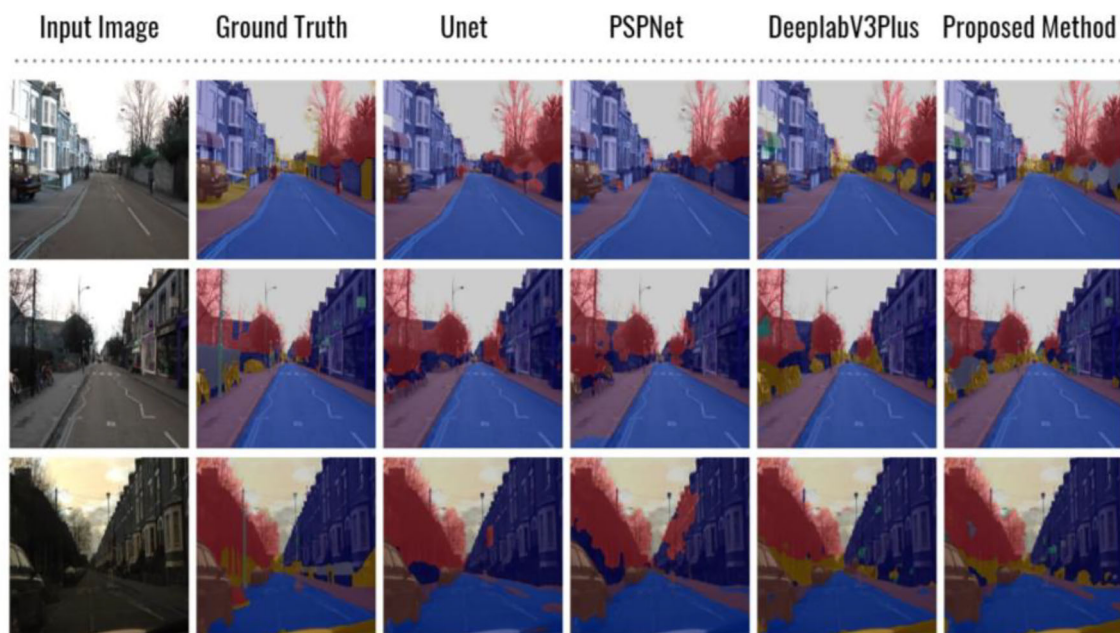
5 Conclusion

In this article, a new semantic segmentation method with higher accuracy performance is proposed so that vehicles can travel autonomously in an urban traffic environment. A new module, Attentional Atrous Feature Pooling (AAFP), has been developed for the proposed method. This module is located between the encoder and decoder in the overall network structure and has been developed to reach more detailed information about objects. In particular, it can obtain multi-scale information and adds the attentional capability to

Table 4 Comparison of segmentation performance between the most advanced methods on the CamVid test set

Method	Input Size	No. of Param(M)	FPS	Class mIoU
ENet [29]	360 × 480	00.36	227.0	51.30
SegNet [15]	360 × 480	29.50	46.00	55.60
FSSNet [30]	360 × 480	00.20	179.0	58.60
ERFNet [31]	360 × 480	02.06	164.0	63.70
DFANet [32]	720 × 960	07.80	120.0	64.70
SwiftNet [33]	720 × 960	12.90	–	65.70
Unet [16]	256 × 256	31.05	35.63	66.41
ICNet [34]	720 × 960	26.50	27.80	67.10
PSPNet [20]	256 × 256	00.58	87.65	67.03
JPANet-M [5]	360 × 480	03.05	256.0	68.29
DeeplabV3+ [17]	256 × 256	11.85	68.36	68.89
Prop. Method without AAFP	256 × 256	09.62	–	70.08
Prop. Method	256 × 256	15.70	64.45	70.59

Bold values highlight the best value

**Fig. 9** Visual comparison of state-of-the-art methods with the CamVid test set

objects large and small. Thanks to this module, the proposed network model was able to realize more effective learning. The training and testing of the proposed network model were carried out with the CamVid dataset. As a result of experimental tests, the average IoU value with the proposed network model was 70.59%, while this value was approximately 2% higher than other state-of-art network models. In addition, the number of parameters and frames per second (FPS) metrics were worth 15.7 million and 64.45, respectively. These values are not a problem given current hardware technologies. However, future studies are planned to further

consider the balance between the accuracy of the method and the amount of consumption of hardware resources.

Author's contribution All authors contributed to the study conception and design. Software coding, methodology, validation and visualization were performed by GD. Review and editing of the study, and data curation was performed by BE.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability The link to all data analyzed during this study is included in this published article.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval This article does not contain any data, or other information from studies or experimentation, with the involvement of human or animal subjects.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Jo K, Kim J, Kim D, Jang C, Sunwoo M (2015) Development of autonomous car - Part II: a case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Trans Industr Electron* 62(8):5119–5132. <https://doi.org/10.1109/TIE.2015.2410258>
- Hafiz AM, Parah SA, Bhat RA (2021) Reinforcement learning applied to machine vision: state of the art. *Int J Multimed Inform Retr* 10(2):71–82. <https://doi.org/10.1007/s13735-021-00209-2>
- Dong G, Yan Y, Shen C, Wang H (2021) Real-time high-performance semantic image segmentation of urban street scenes. *IEEE Trans Intell Transp Syst* 22(6):3258–3274. <https://doi.org/10.1109/TITS.2020.2980426>
- Fan L, Kong H, Wang WC, Yan J (2018) Semantic segmentation with global encoding and dilated decoder in street scenes. *IEEE Access* 6:50333–50343. <https://doi.org/10.1109/ACCESS.2018.2868801>
- Hu X, Jing L, Sehar U (2022) Joint pyramid attention network for real-time semantic segmentation of urban scenes. *Appl Intell* 52(1):580–594. <https://doi.org/10.1007/s10489-021-02446-8>
- Benoughidene A, Titouna F (2022) A novel method for video shot boundary detection using CNN-LSTM approach. *Int J Multimed Inf Retr* 11(4):653–667. <https://doi.org/10.1007/s13735-022-00251-8>
- Cai S, Wang C, Ding J, Yu J, Fan J (2022) FDAM: full-dimension attention module for deep convolutional neural networks. *Int J Multimed Inf Retr* 11(4):599–610. <https://doi.org/10.1007/s13735-022-00248-3>
- Parseh MJ, Rahmanimanesh M, Keshavarzi P, Azimifar Z (2022) Semantic-aware visual scene representation. *Int J Multimed Inf Retr* 11(4):619–638. <https://doi.org/10.1007/s13735-022-00246-5>
- Ilesanmi AE, Ilesanmi T, Idowu OP, Torigian DA, Udupa JK (2022) Organ segmentation from computed tomography images using the 3D convolutional neural network: a systematic review. *Int J Multimed Inf Retr* 11(3):315–331. <https://doi.org/10.1007/s13735-022-00242-9>
- Shotton J, Johnson M, and Cipolla R (2008) Semantic texon forests for image categorization and segmentation. In: 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR. <https://doi.org/10.1109/CVPR.2008.4587503>
- Malik J, Belongie S, Leung T, Shi J (2001) Contour and texture analysis for image segmentation. *Int J Comput Vis* 43(1):7–27. <https://doi.org/10.1023/A:1011174803800>
- Shotton J, Winn J, Rother C, Criminisi A (2009) TextonBoost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int J Comput Vis* 81(1):2–23. <https://doi.org/10.1007/s11263-007-0109-1>
- Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(4):640–651. <https://doi.org/10.1109/TPAMI.2016.2572683>
- Guangzhe Z, Yimeng Z, Ge M, Min Y (2022) Bilateral U-Net semantic segmentation with spatial attention mechanism. *CAAI Trans Intell Technol*. <https://doi.org/10.1049/cit.2.12118>
- Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(12):2481–2495. <https://doi.org/10.1109/TPAMI.2016.2644615>
- Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. *IEEE Access*. https://doi.org/10.1007/978-3-319-24574-4_28
- Chen LC, Zhu Y, Papandreou G, Schroff F, and Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 11211 LNCS, pp 833–851, https://doi.org/10.1007/978-3-030-01234-2_49
- Peng C, Zhang X, Yu G, Luo G, and Sun J (2017) Large kernel matters - Improve semantic segmentation by global convolutional network. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol 2017-Janua, pp 1743–1751. <https://doi.org/10.1109/CVPR.2017.189>
- Wang P et al (2018) Understanding Convolution for Semantic Segmentation. In: *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, vol 2018-Janua, pp 1451–1460. <https://doi.org/10.1109/WACV.2018.00163>
- Zhao H, Shi J, Qi X, Wang X, and Jia J (2017) Pyramid scene parsing network. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol 2017-Janua, pp 6230–6239. <https://doi.org/10.1109/CVPR.2017.660>
- Lin G, Milan A, Shen C, and Reid I (2017) RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5168–5177. <https://doi.org/10.1109/CVPR.2017.549>
- Brostow GJ, Fauqueur J, Cipolla R (2009) Semantic object classes in video: a high-definition ground truth database. *Pattern Recognit Lett* 30(2):88–97. <https://doi.org/10.1016/j.patrec.2008.04.005>
- He K, Zhang X, Ren S, and Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol 2016-Decem, pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Kingma DP and Ba JL (2015) Adam: a method for stochastic optimization. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp 1–15
- Keras, "ReduceLROnPlateau. (2022)
- Dice LR (1945) Measures of the amount of ecologic association between species. *Ecology* 26(3):297–302. <https://doi.org/10.2307/1932409>
- Milletari F, Navab N, and Ahmadi SA (2016) V-Net: Fully convolutional neural networks for volumetric medical image segmentation.

- In: Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016, pp 565–571. <https://doi.org/10.1109/3DV.2016.79>
28. Doğan G, Ergen B (2022) A new mobile convolutional neural network-based approach for pixel-wise road surface crack detection. *Measurement* 195:111119. <https://doi.org/10.1016/j.measurement.2022.111119>
 29. Paszke A, Chaurasia A, Kim S, Culurciello E (2016) ENet: a deep neural network architecture for real-time semantic segmentation. *Comput Vis Pattern Recog*. [arXiv:1606.02147](https://arxiv.org/abs/1606.02147)
 30. Zhang X, Chen Z, Jonathan Wu QM, Cai L, Lu D, Li X (2019) Fast semantic segmentation for scene perception. *IEEE Trans Ind Inform* 15(2):1183–1192. <https://doi.org/10.1109/TII.2018.2849348>
 31. Romera E, Alvarez JM, Bergasa LM, Arroyo R (2018) ERFNet: efficient residual factorized ConvNet for real-time semantic segmentation. *IEEE Trans Intell Transp Syst* 19(1):263–272. <https://doi.org/10.1109/TITS.2017.2750080>
 32. Li H, Xiong P, Fan H, and Sun J (2019) DFANet: Deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 2019-Jun, pp 9514–9523. <https://doi.org/10.1109/CVPR.2019.00975>
 33. Orsic M, Kreso I, Bevandic P, and Segvic S (2019) In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 2019-Jun, pp 12599–12608. <https://doi.org/10.1109/CVPR.2019.01289>
 34. Zhao H, Qi X, Shen X, Shi J, and Jia J (2018) ICNet for Real-time semantic segmentation on high-resolution images. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 11207 LNCS, pp 418–434. https://doi.org/10.1007/978-3-030-01219-9_25

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.