Received: October 29, 2022. Revised: January 12, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Deep Neural Network-based Approach for Accurate Vehicle Counting

Mohamed S. Sawah[1]* **Shereen A. Taie**[1] **Mohamed Hasan Ibrahim**[2] **Shereen A. Hussein**[1]

[1]*Department of Computer Science, Faculty of Computers and Artificial Intelligence, Fayoum University, Egypt*
[2]*Department of Information Systems, Faculty of Computers and Artificial Intelligence, Fayoum University, Egypt*
* Corresponding author's Email: me1900@fayoum.edu.eg

Abstract: In highway management, intelligent vehicle detection and counting are becoming increasingly important as an accurate estimation of traffic density on road congestion reduction. Traffic density estimation is affected by the difficulties of perspective distortion, size change, significant occlusion, and background interference in traffic images. To address the previous issues, this article develops a novel model that enhances the quality of estimating traffic density. The efficientNet fine-tuning architecture is used then, followed by the development of seven dilated convolutional layers to extract the deeper features in the images that maintain the output's resolution to generate a high-quality density map. Finally, the vehicle count will be calculated from the high-quality density map. The experimental results indicate that the suggested approach significantly enhances the accuracy of traffic density estimation compared to the existing ones. It achieves 5.23 as a mean absolute error (MAE) on the TRANCOS dataset.

Keywords: Vehicle counting, Traffic management, Traffic congestion, Surveillance camera, Deep learning.

1. Introduction

It is noticeable that the rate of traffic congestion within cities is increasing, and it is expected to increase in the future. We must work on a large scale to solve these congestions through smart transportation applications and traffic management on the roads, making cities fitter live in. Images are one of the most important sensors to know the extent of the flow of vehicles in large areas. Through networks that contain a large group of cameras placed on roads within cities, it is possible to know the real number of vehicles on the roads using the recent computer vision techniques [1]. Vehicle detection and counting can be used to determine traffic situation, occupancy of on-road lanes, and congestion levels on highways. This data is a major aspect of intelligent transportation systems (ITS). ITS combines a number of technologies into a management system, including automated recognition of license plates, control systems for traffic signals, speed estimation, and incident analysis [2]. Most ITS vehicle counting and detection methods are founded on software or hardware systems. Systems based on hardware have limitations in acquiring precise information on the traffic flow behavior; additionally, they are obtrusive and costly to install and maintain. By contrast, systems based on software have begun to stand out as a low-cost, non-intrusive technique that has been established to be effective, particularly video-based technologies which perform (computer vision) [3].Different strategies are used to accomplish the vehicle counting for detection, but they face difficulties due to the overlap of objects, the instance's size, and the scene's perspective. Given these considerations, this work's major objective is to propose a vehicle counting method to mitigate these difficulties. Appearance-based methods detect employing features with low-level such as color symmetry, texture, edges, and shape, among other characteristics. The scale invariant feature transform (SIFT) is commonly utilized to extract these features [4]. Convolutional neural network methods provide a more advanced detection level by utilizing deep architectures capable of learning complicated information from images [5]. This study introduces a model for counting the vehicles in congested scenes and the traffic density prediction. This prediction has been made by extracting the deeper features in the image representing the different objects at multiple scales and maintaining the output's resolution to generate a high-quality density map. The novelty of this work lies in the ability of receptive field expansion without losing resolution. So, the proposed model can deal with huge differences in scale, perspective, and appearance of vehicles. The structure of the paper is coming as section 2 presents the related works for vehicle counting, section 3 presents the TRANCOS dataset, the proposed model is introduced in section 4, the experimental work and results are explained in section 5, a discussion is introduced in section 6 and the paper is concluded in section 7.

2. Related work

The following sections introduce four types of counting methods: Detection-based methods, regression-based methods, density estimation-based methods, and CNN-based methods.

2.1 Detection-based methods

Detection counting is a supervised strategy in which a previously trained sliding window detector (i.e., a mask that is moved across the whole image) is used for recognizing the objects in the scene. This data is then utilized to calculate the object's number. The mask is trained in monolithic detection to identify the entire object we wish to detect [6]. Finally, the classifier is concerned with object forms in shape-matching detection, such as ellipses [7]. Even though these strategies are straightforward to grasp, they struggle in scenarios containing occlusions. Furthermore, detection algorithms encounter numerous problems, particularly in the task of vehicle countings, for example, changes in occlusions, perspectives, illumination effects, and many others [8]. The author in [9] used image keys and interest points to create the scale-invariant feature transform (SIFT) and the speeded-up robust features (SURF). These extracted features are useful for vehicle detection. The author in [10] used silhouette contours to extract histogram of oriented gradients (HOG) features.The detector (HOG-1) is applied [39] for each training image at multiple scales to collect the detections that will be manually filtered to identify those that contain the correct positive examples, then the (HOG-2) detector is trained using these positive examples so, this way make the detector will be able to train in TRANCOS dataset. The author in [11] produced a collection of Haar-like features, these features, when paired with SVM or AdaBoost, have the potential to dramatically improve the detection model's performance. However, the aforementioned techniques are insufficient and easily influenced by the big rotation in the image. As a result, adjusting the camera angle has a considerable influence on their detection accuracy [12]. Several object detectors based on CNN have recently been proposed, resulting in improved object detection performance. In this context, we highlight two-stage detectors like RCNN [13], Faster-RCNN [14], and Mask-RCNN [15], as well as one-stage detectors like YOLO [16] and SDD [17]. The author in [60] has developed an adaptation technique used to produce a singular patch-based counting regressor capable of counting various object types ,including people, vehicles, cell nuclei and wildlife. The main drawback of this method is that it struggles in scenarios containing occlusions, encounter numerous problems, particularly in the task of vehicle counting, and is affected by the image rotation.

2.2 Regression-based methods

Regression counting is a supervised strategy that attempts to construct a direct (linear or non-linear) mapping from visual features to the number of objects present in the image. It is more resistant to occlusions and distortions of perspective as it is not dependent on a specific model that has previously been developed [6]. By utilizing global image features, regression-based algorithms attempt to detect and count the vehicles (e.g., pixel density and color histogram). To avoid the limitations of the detection-based system, researchers intend to define the problem of vehicle counting as a regression task. They will directly map the image patch's appearance to their related item density maps [18]. These techniques are primarily comprised of two stages: feature extraction and regression modeling. As an input for the counting procedure, the author in [19] provided a set of random regression trees with dense features. The author in [20] presented a cascade regression method for measuring and classifying vehicles based on each segment's low-level data in the foreground. The author in [21] provided a hierarchical classification-based regression (HCR) model that extracts a batch of low-level characteristics from compressed video codec metadata. The author in [22] combined a locally temporal regression method with a spatial regression method. The author in [23] proposed a model for detecting and tracking moving vehicles in traffic scenes using image segmentation and pattern analysis techniques. To encode the object information, a range of local features, such as SIFT [24], HOG [25], LBP [26], and global features, such as texture [27] and gradient [28], were utilized. Gaussian process regression [29], linear regression [30], and ridge regression [31] were used to map the low-level characteristics to a vehicle count. The main drawback of this method is that it is regressed from global features to object count and ignores spatial information.

2.3 Density estimation-based methods

Density estimation for vehicle counting is a supervised strategy that extends the counting by regression approach in various ways. While previous systems dealt satisfactorily with the congestion and occlusion scenes, most of them were regressed from the global features to the number of the objects and neglected any relevant spatial information. In contrast, the author in [32] used a linear mapping between the features of a local patch and the density maps; they first included spatial information in the learning process. As a result, they circumvented the time-consuming process of learning to detect and localize the instances of an individual object by proposing a novel method to estimate the image density that's integral over each image region, showing the number of the objects included inside that region. In [33], a non-linear mapping is presented between density maps and local patch features using a random forest regressor to overcome the problem of linear mapping. They obtained acceptable results by prioritizing crowdedness to account for the major difference in shape and appearance between crowded and un-crowded image patches. The main drawback of this method is that it suffers from the problem of linear mapping.

2.4 CNN-based methods

The task of crowd counting is mostly solved using pixel-based algorithms for counting objects. Hand-crafted feature combinations were employed in [34]. But hand-crafted features are not resistant to occlusion, fluctuation in perspective, vehicle appearance, and scale encountered in actual traffic scenes. Deep neural networks were used to regress the crowd count [35]. Cross-scene crowd counting is accomplished by retrieving scenes with the same perspective as an input test image and patches with the same crowd density as the test image. They next fine-tune a deep network using these extracted patches before testing the input image on the fine-tuned network [36]; however, when the individual CNN columns are not pre-trained, the performance of this design decreases. Because vehicles vary greatly in appearance, the crowd counting deep networks is hard to scale for vehicle counting. Indeed, even inside a single vehicle class, such as cars, there is a great variation, with sedans, hatchbacks, and SUVs falling into the same class. There are a few works that use pixel-based methods to count vehicles. The author in [37] used a perspective-corrected input, a three-stage cascade regression network that deals with different types of vehicles based on their size. Recent work by [38] handles the count of the vehicles in the congested scenes. A multi-scale CNN architecture has been employed to deal with the wide range of viewpoint fluctuations in traffic situations [7]; they trained different custom CNNs for each scale. Using fully connected layers, they combined the feature maps from each trained CNN at a certain scale. As demonstrated by performance across datasets, the performance of counting in [7] is extremely dependent on the levels number in the scale pyramid, revealing a limited ability to generalize. The traffic scene image captured by surveillance cameras shows vehicles with a wide range of visual characteristics such as color, shape, and size. Surveillance cameras have a low resolution and a wide viewing field. While vehicles close to the camera are captured in pretty great detail, vehicles further away are captured in very little detail.

Convolutional neural networks (CNN) like VGG-16 [40], although that the original VGG-16 was trained for image classification, it has been effectively extended to tasks such as semantic pixel-level segmentation. This is due to the extensive sets of filters learned for classification, such as conjunctions in color, edges, and corners. The monolithic CNN employs this variant to compute vehicle counts. As a result, its capacity to depict a complex scene (such as a traffic scene) with tremendous scale and perspective variations is restricted. Additionally, monolithic CNN's employ maximum pooling to merge data following convolution and downsample the spatial resolution [41]. In [42], the image is divided into regions, and a deep neural network is trained with a regressor for each one. In [43], the author introduced a revolutionary counting methodology dubbed gated U-net (GU-Net). To be more precise, incorporate the concept of learnable short-cut links into the U-net design. In deep neural networks, standard short-cut connections between layers are connections that bypass at least one intermediate layer. The author in [44] proposed a model to detect, track, count and

Received: October 29, 2022. Revised: January 12, 2023.

classify the vehicles using blob tracking technologies.The author in [45] proposed a model to count the vehicles in the congested scene using multiple fully convolutional sub-networks to predict the density map for a given static image. In [46], the author introduced an online-update mechanism during training; an online updating technique is employed to update the pseudo ground truth, while a locally constrained regression loss is used to place further constraints on the projected box sizes in a local area by relocating high-level shallow layers features and emphasizing their low-level features. The author in [47] proposed a hierarchical network that improves the low-level features and highlights the high-level features in the image. The author in [48] designed a model that allows any resolution and size for the input image to be mapped into its density map by utilizing filters with varying receptive fields.

A scale pyramid network (SPN) is introduced in [49] that utilizes a common single deep column structure and extracts multi-scale information in the upper levels using the scale pyramid module, while many columns are used to extract multi-scale information from images; the multi-scale information acquired by the multi-column convolutional neural network is aggregated in [50] to increase performance. The author in [51] developed a model based on the idea that a dense region can always be subdivided until the counts of the sub-regions fall within the previously observed closed set. The author in [52] proposed a model capable of counting the direction of movement by integrating time-dependent data. The author in [53] proposed an online vehicle counting model to detect vehicles at crowded intersections using a deep-sort algorithm to perform multi-object tracking. The main drawback of this method is that it is limited ability to generalize. The drawbacks of the four mentioned methods are the motivation of our work. This study introduces a model able to deal with the huge differences in scale, perspectives, and appearance of the vehicles by concentrating on the deeper features in the scene.

However, training multiple deep neural networks requires a huge amount of labeled data. Efficientnet [54] is used as the limitation in vehicles annotation to use its efficiency to predict and generate a high-quality density map to give an accurate vehicle count.

3. Dataset

The TRANCOS dataset [39], which has 1244 images in total, is used for evaluating the proposed model for vehicle counting performance. It is the first dataset to count vehicles in images of traffic jams captured with real-world traffic monitoring cameras. Also, it is frequently used to assess the generalizability of vehicle counting techniques. It was obtained from a selection of public traffic surveillance cameras provided by the Spanish government’s directorate general of traffic (DGT). The cameras picked monitor various motorways in the Madrid area, which are notorious for their intense traffic congestion. Each image has been annotated with a precise number of vehicles and their locations for each image, where 46796 vehicles have been annotated in total. Note that each of the collected images has traffic congestion, spanning a number of diverse scenarios and angles, with varying lighting conditions, varying degrees of crowdedness ,and overlap, even in the same image. The dataset has been divided into a train and test split. The train split consists of 1031 images, while the test split consists of 213 images. The train set is used to train the FCN, while the test set is used to test it and calculate the mean absolute error (MAE) values for the specified test set. Then the deep neural network is initialized with Efficientnet [54]

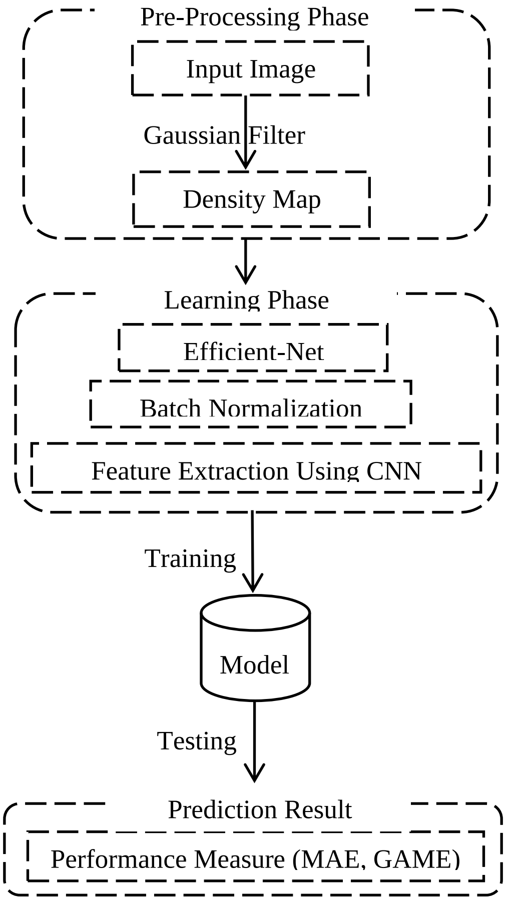

Figure. 1 Model architecture diagram

model weights.

4. Proposed model

4.1 Model architecture

Vehicles are frequently photographed from various viewpoints, resulting in an array of viewpoints and size changes. Vehicles close to the camera are frequently captured in amazing detail. However, when vehicles are not in view of the camera or images are obtained from an aerial perspective. Effective vehicle detection in both of these cases requires the model to work at a highly semantic level concurrently. The proposed design's core idea is to use a deeper CNN depending on EfficientNet CNN models to produce a high-quality density map by capturing the high-level features in images with larger receptive fields. The detailed model that is shown in Fig. 1, is explained in the following subsections.

4.1.1. Pre-processing phase

The primary goal of data pre-processing was to turn the TRANCOS [39] dataset's ground truth into density maps. TRANCOS contains vehicle annotations in the form of a single dot within the vehicle border. The ground truth was generated by creating a vehicle density map that will be used to train the model. The density map is created by applying a Gaussian kernel to blur each vehicle annotation. Normalization is used to ensure that the Gaussian kernel sums to one. The final density map generated for each traffic scene contains the same number of vehicles as the static traffic scene's vehicle count. The scene from the TRANCOS dataset and its derived ground truth is shown in Fig. 2. The derived ground truth is shown graphically as a heat map, with a red area having the vehicle's number higher than the blue areas. A density map of this type simplifies the problem of vehicle counting by providing a coarse signal to the location of the vehicles that play an important role in an accurate prediction.

4.1.2. Learning phase

The learning phase consists of a series of operations as shown in Fig. 3, starting with the EfficientNet fine-tuning, then followed by batch normalization that enables each layer of the network to operate independently of the others. Then the feature extraction using CNN is used, which contains seven dilated convolutional layers to extract the deeper information in images and maintain the resolution of the output to produce a high-quality density map.

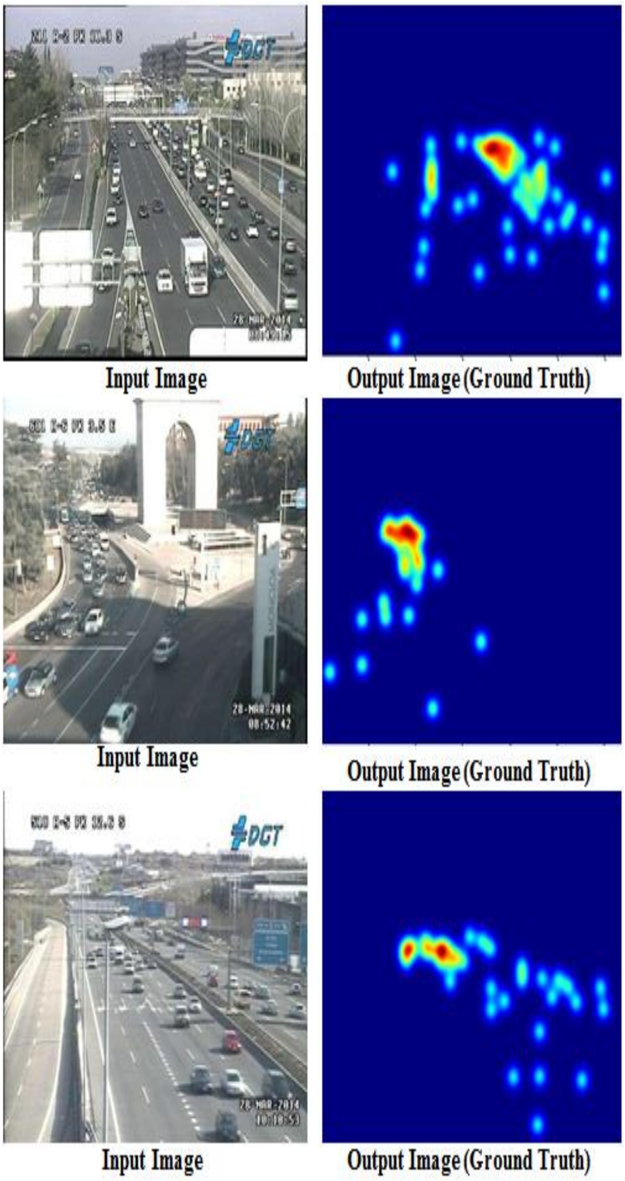
Input Image
Output Image (Ground Truth)
Input Image
Output Image (Ground Truth)
Input Image
Output Image (Ground Truth)

Figure. 2 Shows the traffic scenes from TRANCOS Dataset [39] and its corresponding ground-truth after applying the Gaussian kernel

4.1.2.1. Efficient-net

The author in [54] investigated the relationship between the depth and width of CNN models and devised an effective method for designing CNN models with fewer parameters but higher classification accuracy. They named these EfficientNet CNN models and proposed seven of them in their original paper, named EfficientNetB0 through EfficientNetB7. They demonstrated that when applying EfficientNet to the ImageNet dataset, the EfficientNet CNN models outperform all previous models regarding parameter count and

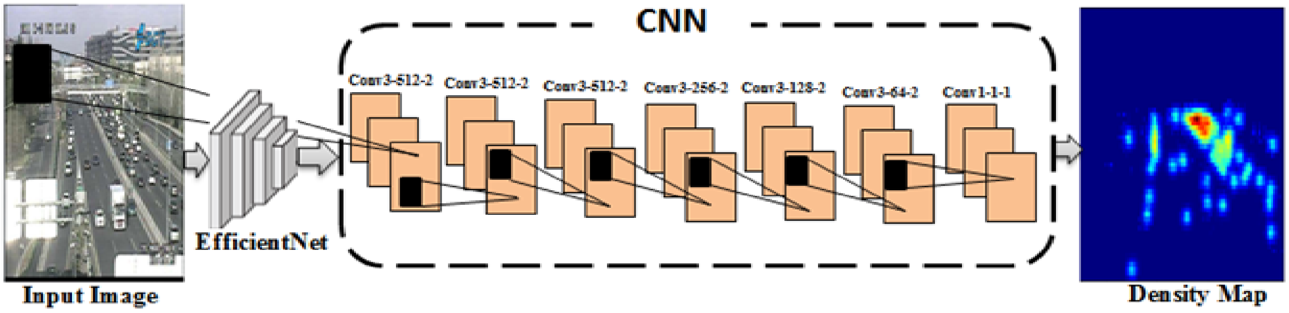

Figure. 3 The model framework. The parameter of the convolutional layers is denoted as "Conv-(kernel size)-(number of filters)-(dilation rate)"

Top-1 accuracy. The EfficientNet family of algorithms is based on a novel technique for scaling CNN models. It makes use of a simple, extremely powerful compound coefficient. Unlike existing approaches that scale network parameters such as depth, width, and resolution, EfficientNet scales each dimension uniformly using a predefined set of scaling factors. Scaling individual dimensions enhances model performance, but balancing all network dimensions to available resources significantly improves the model's overall performance. The efficient technique based on the EfficientNet-B5 CNN model is provided in this study. This variant of the EfficientNet-B5 has been chosen specifically due to its excellent trade-off between processing resources and accuracy [55-56]. Using the EfficientNet network's representational power, the proposed model fine-tunes its filters to solve the problem of vehicle counts more effectively.

4.1.2.2. Batch normalization

Batch normalization is a layer that enables each layer of the network to operate independently of the others. It is placed after the convolution layers to normalize the output of the previous layers, learning becomes more efficient, and it may also be utilized as a regularization technique to avoid model overfitting. Also, it is used to standardize the model's inputs and outputs. It is utilized at several points within the model's layers.

4.1.2.3. Feature extraction using CNN

After the EfficientNet fine-tuning, a seven dilated convolutional layers, according to the works [57-59], will be mentioned in the following subsection as the back-end to extract the deeper information in images and maintain the resolution of the output to produce a high-quality density map as shown in Fig. 2.

4.1.2.4. Dilated convolution

The dilated convolutional layer is a fundamental component of the proposed model architecture. A dilated convolution in two dimensions can be formulated as:

$$y(m,n) = \sum_{i=1}^{M} \sum_{j=1}^{N} x(m+r\times i, n+r\times j)w(i,j) \qquad (1)$$

$y(m,n)$ is the dilation convolution output from the input $x(m,n)$ and a filter $w(i,j)$ with length M and width N, and the dilation rate is r. If $r=1$, A dilated convolution becomes a normal convolution. The dilation convolutional layers have been shown to significantly improve accuracy in segmentation tasks [57-59] and are a viable alternative to pooling layers. While pooling layers are frequently used to maintain invariance and control overfitting, they significantly diminish spatial resolution, hence obliterating the spatial information contained in feature maps. Dilated convolution is a superior alternative, as it alternates the pooling and convolutional layers using sparse kernels .

This property increases the receptive field without increasing the computation or number of parameters required (for example, adding more convolutional layers increases the receptive field but introduces additional operations). A small-size kernel with $k\ x\ k$ a filter in a dilation convolution is widened to $k+(k-1)(r-1)$ with dilated stride r. As a result, it enables flexible aggregation of multi-scale contextual data while maintaining the same resolution. As a result, dilated convolution demonstrates different advantages over the scheme of convolution, pooling, and deconvolution to maintain the resolution of the feature map.

4.1.2.5. Training details

The model is trained using an Adadelta

optimizer with a base learning rate ($\alpha = 0.05$ based on trials and error operations). Throughout the training, a consistent learning rate was maintained, and consistent training was maintained across architectures. In addition, the Euclidean distance is used to assess the difference between the ground truth and the estimated density map, which is similar to the method utilized by [48]. Here is the formula for the loss function:

$$L(\Theta) = \frac{1}{2N}\sum_{n=1}^{N} \|e_n(\Theta) - gt_n\|_2^2 \qquad (2)$$

Where Θ is the set of learnable parameters in the model, N is the number of the training image and $e_n(\Theta)$ is the estimated object count generated by the model, which is parameterized with Θ and gt_n is the provided ground truth for image n.

4.1.3. Prediction result

4.1.3.1. Performance measures

The model is tested after the training process using the following measures:

- The mean absolute error (MAE).

The mean absolute error (MAE) is used as the performance metric for the proposed model, which is formulated as:

$$MAE = \frac{1}{N}\cdot\sum_{n=1}^{N} |e_n - gt_n| \qquad (3)$$

- The grid average mean absolute error (GAME).

To provide a more accurate evaluation, the grid average mean absolute error (GAME) measure [39] is used, which provides an evaluation metric that takes both the object count and the estimated location of the objects into account, which is formulated as:

$$GAME(L) = \frac{1}{N}\cdot\sum_{n=1}^{N}\left(\sum_{l=1}^{4^L} \left|e_n^l - gt_n^l\right|\right) \qquad (4)$$

Where e_n^l is the count estimated in a region l of an image n, and gt_n^l is the image's ground truth for the same area. The greater the value L, the more restrictive the GAME metric. Notably, when $L = 0$ the MAE can be obtained as a particularization of the GAME.

Table 1. Show the results on TRANCOS dataset

No.of epochs	MAE =G(0)	G(1)	G(2)	G(3)
50	9.40	11.07	12.13	14.05
60	8.99	10.56	11.60	13.40
70	9.70	11.14	12.14	13.83
80	8.76	10.55	11.64	13.59
90	8.84	10.73	11.75	13.70
100	8.70	10.26	11.43	13.43
110	8.58	10.53	11.43	13.40
120	9.06	10.94	11.91	13.77
130	8.88	10.44	11.53	13.63
140	5.79	5.94	6.05	6.82
150	8.68	10.36	11.39	13.33
160	5.25	5.39	5.53	6.36
170	5.23	5.39	5.52	6.40
180	8.90	10.62	11.66	13.54
190	8.92	10.23	11.34	13.15
200	8.87	10.37	11.37	13.34
210	8.45	10.18	11.19	13.05
220	8.97	11.80	13.10	15.27
230	8.68	10.33	11.35	13.25
240	8.93	10.27	11.28	13.17
250	8.54	10.22	11.23	13.17
260	9.30	10.77	11.75	13.49
270	8.70	10.46	11.43	13.34
280	8.66	10.21	11.15	13.01
290	9.24	10.95	11.54	13.30
300	8.54	10.22	11.23	13.17

5. Experimental work and results

All experiments are conducted on the TRANCOS dataset. The density map has been utilized as the ground truth. The model is trained using an Adadelta optimizer with a base learning rate ($\alpha = 0.05$ based on trials and error operations).

Throughout the training, a consistent learning rate was maintained, and consistent training was maintained across architectures.

The experimental results indicate that the proposed model achives a promising results, indicating that it can perform well in the extremely overlapping traffic congestion scenario.

As shown in Table 1, the model's results using EfficientNetB5 fine-tuning at a learning rate equal to 0.05 and Adadelta optimizer for a different number of epochs starting with 50 to ending with 300 epochs show that the best result (i.e., the smallest MAE=5.23) at 170 epochs to overcome the overfitting problem.

As shown in Table 2, Reg. Forest [19] has the highest mean absolute error of 17.77 and the highest GAME values on the TRANCOS. While theperformance of the latest deep learning methods, such as CCNN [18], Tra-Count [45], AMDCN [50], CoarseAdapt [53], and Adv. Dmap [54] is superior to that of the Reg. Forest [19]. The proposed model achives a promising results with mean absolute error (MAE = G(0) = 5.23) and (G(1)= 5.39, G(2)= 5.52 and G(3)= 6.40) .

(___) in Table 2 idicates that the reference used the mean absolute error metric only to evaluate their work and did not use GAME(1), GAME(2) and GAME(3) metrics.

In Tables 1 and 2, G(0) = GAME(0), G(1) =GAME(1), G(2)=GAME(2) and G(3)=GAME(3).

In addition, in Fig. 4, different snapshots of the applied approach refer to the efficiency of detecting vehicle density in frames in the proposed model, as the third column shows the predicted count of the vehicles that are very close to the original count in column two. It also confirmed the sufficiency of the model’s robustness and accuracy even in the presence of severe barrier scenarios.

Table 2. Show the comparison results

Methods	**Year**	**G(0)**	**G(1)**	**G(2)**	**G(3)**
MESA [32]	2010	13.76	16.72	20.72	24.36
Reg. Forest [19]	2012	17.77	20.14	23.65	25.99
HOG-2[39]	2015	13.29	18.05	23.65	28.41
CCNN [18]	2016	12.49	16.58	20.02	22.41
Hydra-2s[18]		11.41	16.36	20.89	23.67
Hydra-3s[18]		10.99	13.75	16.69	19.32
Hydra-4s[18]		12.92	15.54	18.45	20.96
MCNN[48]	2016	11.05	____	____	____
Tra-Count[45]	2016	8.12	——	——	——
FCN-ST[42]	2017	5.47	____	____	____
AMDCN[50]	2018	9.77	13.16	15.00	15.87
NCNN[52]	2019	10.79	____	____	____
Skip-Network[47]	2020	7.38	——	——	——
Proposed Model	2022	5.23	5.39	5.52	6.40

6. Disscussion

The proposed model consists of several stages, in the pre-processing stage, the goal is to turn the TRANCOS dataset's ground truth into density maps that will be used to train the model. TRANCOS contains vehicle annotations as a single dot within the vehicle border. The ground truth was generated by creating a vehicle density map. The density map is created by applying a Gaussian kernel to blur each vehicle annotation. Normalization is used to ensure that the Gaussian kernel sums to one. The final density map generated for each traffic scene contains the same number of vehicles as the static traffic scene's vehicle count that will be used to train the model. In the learning phase, a series of operations starting with the EfficientNet fine-tuning, then followed by batch normalization that enables each layer of the network to operate independently of the others. Then the feature extraction using CNN is used, which contains seven dilated convolutional layers to extract the deeper information in images and maintain the resolution of the output to produce a high-quality density map. The proposed method is evaluated on TRANCOS dataset. The experiments included two performance measures. Firstly, the mean absolute error (MAE) is used as the performance metric. Secondly, the grid average mean absolute error (GAME) measure is used, which provides an evaluation metric that takes both the object count and the estimated location of the objects into account. After training 300 epochs, we found the lowest MAE using the evaluation images at epoch 170 and later epochs seem to be overfitted. The proposed model achieve a mean absolute error (MAE = GAME(0) = 5.23) and (GAME(1)= 5.39, GAME(2)= 5.52 and GAME(3)= 6.40) which is significantly achives a promising results.

7. Conclusion

This paper introduces a model for counting the vehicles in the congested scenes and the traffic

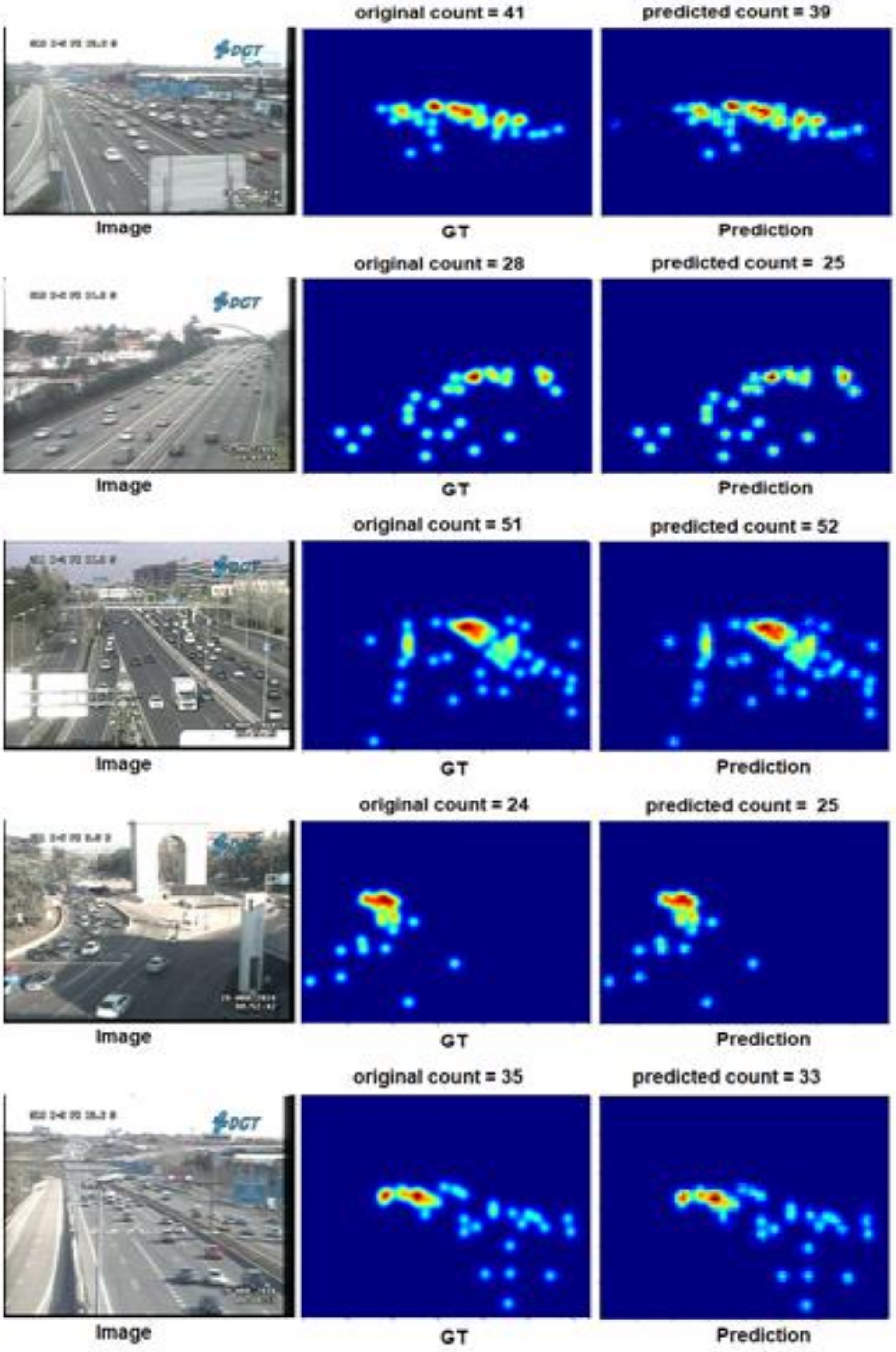

Figure. 4 The first column shows images from TRANCOS Dataset [39]. The second column shows the ground truth (GT) for each image and the original count of vehicles in the image. The third column shows the generated density map and the predicted count of the vehicles in the image by our model

density prediction. This prediction has been done by extracting the deeper features in the image that represents the different objects at multiple scales and maintaining the output's resolution to generate a high-quality density map. EfficientNet is used in the proposed model to scale each dimension uniformly using a predefined set of scaling factors. Scaling individual dimensions enhances model performance, but balancing all network dimensions in relation to available resources significantly improves the model's overall performance. The dilated convolutional layers are used in the proposed model to aggregate the multi-scale contextual information in the congested scenes. It is also used to alternate the pooling and convolutional layers using sparse kernels. The proposed model performance was evaluated in two metrics. The first metric is the mean absolute error (MAE). While this metric appears reasonable for constructing a comparison, the investigations show that it frequently masks incorrect estimations. The reason for this is that the MAE does not take into account the location of the estimations in the images. The second metric is the grid average mean absolute error (GAME) which provides an evaluation metric that takes both the object count and the estimated location of the objects into account. The experimental results indicate the proposed model achives a promising results in the two metrics. The novelty of this work lies in the ability of receptive field expansion without losing resolution. So, the proposed model can deal with huge differences in scale, perspective, and appearance of vehicles. The proposed model achieved MAE = 5.23 on the challenging TRANCOS dataset. Finally, this work demonstrated the value of the high-quality density map for vehicle counting calculation in high performance.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization of this paper, S. Taie, S. Ahmed and M. Sawah; methodology, S. Taie, S. Ahmed, and M. Sawah; the software, S. Ahmed and M. Sawah; writing (original draft), M. Sawah; review and editing, S. Taie, Mohamed H. Ibrahim, S. Ahmed and M. Sawah.

Acknowledgments

The authors thank the editors and the anonymous reviewers for their valuable suggestions.

References

[1] L. Ciampi, C. Gennaro, F. Carrara, F. Falchi, C. Vairo, and G. Amato, “Multi-Camera Vehicle Counting using Edge-AI”, *Expert Systems with Applications*, Vol. 207, 2022.

[2] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, “Data-Driven Intelligent Transportation Systems: A Survey”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, pp. 1624–1639, 2011.

[3] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, “Vision-based Vehicle Detection and Counting System using Deep Learning in Highway Scenes”, *European Transport Research Review*, Vol. 11, No. 1, pp. 1-16, 2019.

[4] L. Juan and O. Gwun, “A Comparison of Sift, Pca-Sift and Surf”, *International Journal of Image Processing (IJIP)*, Vol. 3, No. 4, pp. 143–152, 2009.

[5] Z. Zhao, P. Zheng, S. Xu, and X. Wu, “Object Detection With Deep Learning: A Review”, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 30, No. 11, pp. 3212-3232, 2019.

[6] L. Ciampi, G. Amato, F. Falchi, C. Gennaro, and F. Rabitti, “Counting Vehicles with Cameras” , In: *Proc. of the 26th Italian Symposium on Advanced Database Systems*, pp. 24-27, 2018.

[7] T. Zhao, R. Nevatia, and B. Wu, “Segmentation and Tracking of Multiple Humans in Crowded Environments”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 7, pp. 1198-1211, 2008.

[8] J. Majin, Y. Valencia, M. Stivanello, M. Stemmer, and J. Salazar, “a Novel Deep Learning-Based Method for Detection and Counting of Vehicles in Urban Traffic Surveillance Systems”, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 43, pp. 793–800, 2021.

[9] H. Bay, T. Tuytelaars, and L. Gool, “Surf: Speeded up robust features”, In: *Proc. of European Conference on Computer Vision*, pp. 404–417, 2006.

[10] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection”, In: *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, 2005.

[11] R. Lienhart and J. Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection”, In: *Proc. of International Conference on Image Processing*, pp. 1–1, 2002.

[12] D. Dinh, H. Nguyen, H. Thai, and K. Le, “Towards AI-Based Traffic Counting System with Edge Computing”, *Journal of Advanced Transportation*, pp. 1–15, 2021.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, In: *Proc. of the IEEE Conference on Computer*

Vision and Pattern Recognition, pp. 580– 587, 2014.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, *Advances in Neural Information Processing Systems*, Vol. 28, 2015.

[15] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object Detection”, In: *Proc. of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-time Object Detection”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A.Berg, “Ssd: Single Shot Multibox Detector”, In: *Proc. of European Conference on Computer Vision*, pp. 21–37, 2016.

[18] D. Rubio and R. Sastre, “Towards Perspective-free Object Counting with Deep Learning”, In: *Proc. of European Conference on Computer Vision*, pp. 615–629, 2016.

[19] L. Fiaschi, R. Nair, U. Koethe, and F. Hamprecht, “Learning to Count with Regression Forest and Structured Labels”, In: *Proc. of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 2685–2688, 2012.

[20] M. Liang, X. Huang, C. Chen, X. Chen, and A. Tokuta, “Counting and Classification of Highway Vehicles by Regression Analysis”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 5, pp. 2878–2888, 2015.

[21] X. Liu, Z. Wang, J. Feng, and H. Xi, “Highway vehicle counting in compressed domain”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3016–3024, 2016.

[22] Z. Wang, X. Liu, J. Feng, J. Yang, and H. Xi, “Compressed-Domain Highway Vehicle Counting by Spatial and Temporal Regression”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 29, No. 1, pp. 263–274, 2017.

[23] Y. Chen, B. Wu, H. Huang, and C. Fan, “A Real-Time Vision System for Nighttime Vehicle Detection and Traffic Surveillance”, *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 5, pp. 2030–2044, 2011.

[24] K. Tota and H. Idrees, “Counting in Dense Crowds Using Deep Features”, In: *Proc. of CRCV*, pp. 1–4, 2015.

[25] Z. Ma and A. Chan, “Crossing the line: Crowd Counting by Integer Programming with Local Features”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 11, No. 5, pp. 2539–2546, 2013.

[26] Z. Wang, H. Liu, Y. Qian, and T. Xu, “Crowd Density Estimation Based on Local Binary Pattern Co-occurrence Matrix”, In: *Proc. of IEEE International Conference on Multimedia and Expo Workshops*, Vol. 11, No. 5, pp. 372–377, 2012.

[27] S. Ghidoni, G. Cielniak, and E. Menegatti, “Texture-based Crowd Detection and Localisation”, *Intelligent Autonomous Systems*, pp. 725–736, 2013.

[28] J. Balbin, R. Garcia, K. Fernandez, N. Golosinda, K. Magpayo, K. Denise, and R. Velasco, “Crowd Counting System by Facial Recognition using Histogram of Oriented Gradients, Completed Local Binary Pattern, Grey-Level Co-occurrence Matrix, and Unmanned Aerial Vehicle”, In: *Proc. of SPIE 10828, Third International Workshop on Pattern Recognition*, pp. 108280Y, 2018.

[29] A. Chan and N. Vasconcelos, “Counting People With Low-Level Features and Bayesian Regression”, *IEEE Transactions on Image Processing*, Vol. 21, No. 4, pp. 2160– 2177, 2011.

[30] X. Huang, Y. Zou, and Y. Wang, “Cost-Sensitive Sparse Linear Regression for Crowd Counting with Imbalanced Training Data”, In: *Proc. of IEEE International Conference on Multimedia and expo (ICME)*, pp. 1–6, 2016.

[31] K. Chen, C. Loy, S. Gong, and T. Xiang, “Feature Mining for Localised Crowd Counting”, In*: Proc. of Bmvc*, Vol. 1, No. 2, p. 3, 2012.

[32] V. Lempitsky and A. Zisserman, “Learning to Count Objects in Images”, *Advances in Neural Information Processing Systems*, Vol. 23, 2010.

[33] V. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, “Count forest: Co-voting Uncertain Number of Targets using Random Forest for Crowd Density Estimation”, In: *Proc. of the IEEE International Conference on Computer Vision*, pp. 3253–3261, 2015.

[34] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, “Multi-Source Multi-ScaleCounting in Extremely Dense Crowd Images”, In: *Proc. of the IEEE Conference on Computer vision and Pattern Recognition*, pp. 2547–2554, 2013.

[35] C. Wang, H. Zhang, L. Yang, S. Liu, and X.

Cao, “Deep People Counting in Extremely Dense Crowds”, In: *Proc. of the 23rd ACM International Conference on Multimedia*, pp. 1299–1302, 2015.

[36] C. Zhang, H. Li, X. Wang, and X. Yang, “Cross-Scene Crowd Counting via Deep Convolutional Neural Networks”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 833–841, 2015.

[37] M. Liang, X. Huang, C. Chen, X. Chen, and A. Tokuta, “Counting and Classification of Highway Vehicles by Regression Analysis”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 5, pp. 2878–2888, 2015.

[38] X. Liu, Z. Wang, J. Feng, and H. Xi, “Highway Vehicle Counting in Compressed Domain”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3016–3024, 2016.

[39] R. Olmedo, B. Jimenez, R. Satre, S. Bascon, and D. Rubio, “Extremely Overlapping Vehicle Counting”, In: *Proc. of Iberian Conference on Pattern Recognition and Image Analysis*, pp. 423–431, 2015.

[40] M. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks”, In: *Proc. of European Conference on Computer Vision*, pp. 818–833, 2014.

[41] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, In: *Proc. of the Computing Research Repository*, 2015.

[42] S. Zhang, G. Wu, J. Costeira, and J. Moura, “Understanding Traffic Density from Large-Scale web Camera Data”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5898–5907, 2020.

[43] D. Rubio, M. Niepert, and R.Sastre, “Learning short-cut connections for object counting”, In: *Proc. of the 29th British Machine Vision Conference*, 2018.

[44] S. Veni, A. Hiremath, M. Patil, M. Shinde, and A. Teli, “Video-Based Detection, Counting and Classification of Vehicles Using OpenCV”, In: *Proc. of the International Conference on IoT Based Control Networks and Intelligent Systems*, 2020.

[45] S. Surya and V. Babu, “TraCount: a Deep Convolutional Neural Network for Highly Overlapping Vehicle Counting”, In: *Proc. of the Tenth Indian Conference on Computer Vision*, pp. 1–6, 2016.

[46] Y. Liu, M. Shi, Q. Zhao, and X. Wang, “Point in, Box Out: Beyond Counting Persons in Crowds”, In: *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6469–6478, 2019.

[47] S. Sooksatra, T. Kondo, P. Bunnun, and A. Yoshitaka, “Redesigned Skip-network for Crowd Counting with Dilated Convolution and Backward Connection”, *Journal of Imaging*, Vol. 6, No. 5, 2020.

[48]Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-Image Crowd Counting via Multi-column Convolutional Neural Network”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern recognition*, Vol. 11, No. 5, pp. 589–597, 2016.

[49] X. Chen, Y. Bin, N. Sang, and C. Gao, “Scale pyramid network for crowd counting,” In: *Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1941–1950, 2019.

[50] D. Deb and J. Ventura, “An Aggregated Multicolumn Dilated Convolution Network for Perspective-Free Counting”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 195–204, 2018.

[51] H. Xiong, H. Lu, C. Liu, L. Liu, Z. Cao, and C. Shen, “From Open Set to Closed Set:Counting Objects by Spatial Divide-and-Conquer”, In: *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 8362–8371, 2019.

[52] O. Urbann and J. Stenzel, “A convolutional Neural Network that Self-Contained Counts”, *Journal of Image and Graphics*, Vol. 7, No. 4, pp. 112–116, 2019.

[53] J. Lu, M. Xia, X. Gao, X. Yang, T. Tao, H. Meng, W. Zhang, X. Tan, Y. Shi, G. Li, and E. Ding, “Robust and Online Vehicle Counting at Crowded Intersections”, In: *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4002-4008, 2021.

[54] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks”, In: *Proc. of International Conference on Machine Learning*, pp. 6105–6114, 2019.

[55] M. Maswood, T. Hussain, M. Khan, M. Islam, and A. Alharbi, “CNN Based Detection of the Severity of Diabetic Retinopathy from the Fundus Photography using EfficientNet-B5”, In: *Proc. of 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0147–0150, 2020.

[56] K. Li, Z. Shaikh, A. Khan, and A. Laghari, “Multiclass Skin Cancer Classification using EfficientNets--A First Step Towards Preventing Skin Cancer”, *Neuroscience Informatics*, Vol. 2, 2021.

[57] F. Yu and V. Koltun, “Multi-Scale Context Aggregation by Dilated Convolutions”, *arXiv Preprint arXiv:1511.07122*, 2015.

[58] L. Chen, G. Papandreou, L. Kokkinos, K. Murphy, and A. Yuille, “Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully connected Crfs”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 4, pp. 834–848, 2017.

[59] L. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation”, *arXiv Preprint arXiv: 1706.05587,* 2017.

[60] M. Marsden, K. Guinness, S. Little, C. Keogh, and N. Connor, “People,Penguins and Petri Dishes: Adapting Object Counting Models to New Visual Domains and Object Types without Forgetting”, In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8070-8079, 2018.