

A Comparative Analysis Employing Adaptive Layers of RCNN Technique and Transfer Learning Pre-Trained Networks



Maha A. Joodi^{*}, Fatin E.M. Al-Obaidi[†], Ali A.D. Al-Zuky[‡]

Department of Physics, College of Science, Mustansiriyah University, Baghdad 10011, Iraq

Corresponding Author Email: mahaaziz@uomustansiriyah.edu.iq

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ria.380408>

ABSTRACT

Received: 29 December 2023

Revised: 1 April 2024

Accepted: 11 June 2024

Available online: 23 August 2024

Keywords:

precision, recall, region-based-convolution neural network (RCNN)

The study aimed to classify two classes of vehicles, Tuktuk and Motorcycle, using a modified RCNN model. The MAjN_IRAQ Dataset, created from a camera system in Baghdad city, was used for training, detection, and classification of some vehicles to allow them to enter some crowded streets of Baghdad and to prevent others from entering the same streets. New layers were added and the number and size of filters were changed, which led to improve the process of training, detection and classification of vehicles with high accuracy, which leads to improving the proposed model's performance. The results showed that the modified RCNN model performed better when trained for 80 epochs. It improved performance measures such as precision, recall, and F1 score measure. The model was compared to other transfer learning methods (Alex Net, VGG16, and VGG19) and showed superior results for the Tuktuk class. The training and testing time for the proposed RCNN-modified model was also lower compared to the other models. At 80 epochs, the precision for the Tuktuk class was approximately 0.94, while for the Motorcycle class, it was approximately 0.89. The TPR was higher for the Tuktuk class at approximately 0.93, while the lower value was approximately 0.84 for VGG16. When the VGG16 model was used, the F1 score was better in the Motorcycle category (about 0.95) but worse in the Tuktuk category (0.86%). Both the suggested RCNN-modified model and the Alex Net model worked well in a reasonable amount of time.

1. INTRODUCTION

The adaptability of RCNN is resulting in an increasing recognition of its significance in vehicle identification. This technology facilitates accurate identification and surveillance of cars, hence improving traffic management, autonomous driving technologies, and security monitoring. By identifying distinguishing traits and training the network with a large dataset through the analysis of image or video frames, RCNN can identify vehicles. Among the several applications that have found success with this approach are those related to transportation, security, and smart cities. Nonetheless, while employing RCNN for vehicle detection, researchers encounter a few challenges. One challenge is the complexity of the network architecture, which requires substantial computational resources and training time. Furthermore, the detection accuracy is influenced by the variations in vehicle appearances that result from lighting conditions, occlusions, and changes in viewpoint [1-3].

Two vehicle-finding deep learning models were compared by Espinosa et al. [4]. An urban video sequence was analyzed and compared with Faster R-CNN and Alex Net [5]. To assess the effectiveness of the detections, timeframes, and failure rates required to finish the detection task, multiple tests were conducted. Whereas Alex Net required over 100 ms per frame, the Faster R-CNN model achieved near real-time (40 ms per

frame).

The outcomes enable the drawing of significant conclusions about the strategies and architectures utilized to put such a network into practice for the purpose of video detection, stimulating more study on this subject.

Arinaldi et al. [6] presented a computer vision-based traffic video analysis system. Policy makers and regulators can use the system to collect important statistics automatically. These data comprise lane usage monitoring, vehicle type categorization, vehicle counting, and video-based speed estimation. In such a system, vehicles were detected and classified using video footage of traffic. For this purpose, two models were implemented. First was a mixture of Gaussian (MoG) background subtraction + a support vector machine (SVM) system. The second method used Faster RCNN, a deep learning architecture that has become popular recently to detect subjects in images. In experiments, when it comes to detecting cars that are stationary, overlapping, or operating at night, faster RCNN performs better than MoG. In addition, faster RCNN is more effective than SVM when it comes to determining the type of vehicle-based on its appearance.

It was used by Murugan et al. [7], along with removal and box filter-based background estimation. A box filter-based background estimation was implemented to mitigate the rapid fluctuations that resulted from the movement of automobiles. Next, by examining the pixel-by-pixel differences between the

input frames and the estimated background, moving cars were identified. After detecting variant vehicles, a recognition phase was performed in order to classify them. To recognize vehicles with region proposals, the deep learning framework RCNN was employed. Computational multiplicity was decreased because RCNN has region proposals. A number of metrics, such as sensitivity, accuracy, precision, and specificity, were evaluated in order to determine the proficiency of the region proposals used in the RCNN that enabled the model to achieve 91.3% recognition accuracy for variant types of vehicles.

However, Nguyen [8] presented a better way to find fast vehicles using Faster R-CNN. First, the foundational convolution layer of the Faster R-CNN was constructed using the MobileNet architecture [5]. Afterwards, the original Faster R-CNN's NMS algorithm that followed the region proposal network was replaced with the soft-NMS algorithm to address the problem of repeated proposals. As a next step, the proposals were adjusted to the specified size while maintaining key contextual information using a context-aware RoI pooling layer. To build the classifier in the final stage of the Faster R-CNN framework, the MobileNet architecture makes use of the structure of depthwise separable convolution. Each detected vehicle's bounding box is adjusted, and suggestions are classified by this classifier. Their proposed solution proved to be both faster and more accurate than the original Faster R-CNN, after testing it on the LSVH and KITTI car datasets. When compared to the original Faster R-CNN framework, their model showed a 24.5% improvement in LSVH performance and a 4% improvement in KITTI performance.

A 2019 solution was developed by Grents et al. [9] using a SORT tracker and a two-stage Faster R-CNN detector. There were over 52,000 objects in 750 video frames from various environments used to train and test the detectors. Their system can sort, count, and calculate vehicle speeds with an average absolute percentage error of no more than 22%, according to their experiments. Faruque et al. [10] used the Faster R-CNN and the YOLO to classify vehicles. For the YOLO deep learning and Faster R-CNN methods, the initial step was to manually create three training data sets from two low-quality videos. Additional newly recorded videos that were not used in training were also used to assess how well deep learning methods classified vehicles. The comparative study focuses on evaluating the accuracy of vehicle classification, as well as the time required for testing and training, and the ability of deep learning techniques to generalize. Research findings indicate that the YOLO deep learning methodology demonstrates considerably faster speed than the Faster R-CNN deep learning method. These investigations also verified the feasibility of utilizing deep learning techniques for the categorization of automobiles in movies.

Htet and Sein [11] demonstrated the procedure of categorizing and enumerating motor vehicles for an event system. First, the event video streaming camera's images were featured, extracted, segmented. They were then presented to achieve improved optimization using the dataset of the modified Stanford car and the new Myanmar cars dataset, as well as the Deep Neural Learning Fast R-CNN method with the optimization of hyper-parameter. With the created Myanmar cars dataset, the hyper-parameter and Fast RCNN classifier showed state-of-the-art performance in vehicle counting and classification using real-time video streaming of real-life events.

Algiriyage et al. [12], as the first stage of a larger project, focused on choosing an acceptable object detection model for counting and identifying cars from closed-circuit television

(CCTV) images and next assessing the flow of traffic. The three widely used object detection models, mask R-CNN, faster R-CNN, and YOLOv3, were first assessed for performance accuracy. The outcomes of their experiment showed that, in comparison to the other two models, YOLOv3 could detect objects extremely quickly. They also showed that YOLOv3 had the best accuracy in the same experiment.

There are two primary phases to the suggested strategy: detection and multiple-object tracking [13]. First, a Faster R-CNN model was employed to identify and categorize the vehicles into the following classes: cars, motorbikes, buses, trucks, and rudimentary vehicles. The model can operate reliably in real-time with an accuracy of more than 86 percent, according to the results.

Sharma et al. [14] employed an automated technique using a modified RCNN for video analysis in order to identify vehicles. Vehicle identification in a certain frame was examined in the traffic footage gathered by CCTV cameras placed on the roadways. The characteristics were extracted using the pre-trained Google Net. The RCNN utilized these characteristics to identify the vehicle. Utilizing a probability score calculated by utilizing object intersection (IoU), the vehicles were recognized. The detected vehicles are categorized into ten distinct vehicle classifications. Several different network models were utilized to test and contrast the approach, which demonstrated more accuracy.

Ghasemi Darehnaei et al. [15] presented the SI-EDTL, or swarm intelligence ensemble deep transfer learning, for the purpose of detecting numerous vehicles in photos taken by unmanned aerial vehicles (UAVs). The region proposal network (RPN) is utilized to extract a set of region proposals in this approach, which is based on Faster regional-based convolutional neural networks (Faster R-CNN). Thereafter, CNN is utilized to categorize areas by mining these windows' highly descriptive properties. Consequently, a UAV dataset is used to train 15 distinct base learners via deep transfer learning in order to categorize the area suggestions into various automobiles (car, truck, van, and bus). By using a weighted average aggregation, they integrate these 15 base learners into four categories of vehicles or none at all (background). To find the optimal trade-off between overall accuracy, recall, and precision, the whale optimization algorithm is used to modify the hyperparameters of the ensemble model. Using MATLAB R2020b's parallel processing feature, their SI-EDTL model was effectively constructed. The SI-EDTL model outperforms other methods, as shown by experimental findings on the AU-AIR dataset of UAV photos.

Djenouri et al. [16] introduced a better regional convolution neural network, and the vehicle identification problem is investigated. The SIFT extractor is used to first gather the vehicle data (set of pictures), from which the noise (set of outlier photos) is extracted. The vehicles are subsequently detected using the area convolution neural network. An evolutionary computation-based hyper-parameter optimization technique is suggesting to adjust the deep learning framework's parameters.

Alam et al. [17] introduced a novel approach to improve the performance of detection findings in rapid vehicle recognition. They utilized multiscale feature maps from CNN and input pictures with varying resolutions to adapt the base network, resulting in enhanced detection efficiency and processing time. Their proposed methodology, built upon Faster R-CNN, outperformed earlier versions of Faster R-CNN models. They used four different base networks (Modified Vgg16, MobilenetV3, ResNet50, and ResNet101) as feature extractors

and found that their modified Vgg16 model achieved better accuracy and faster testing time in recognizing automobile categories in their custom detection dataset.

This paper presents a vehicle classification model which is capable of recognizing two classes of vehicles (tuktuk and motorcycle). The proposed method of detecting and recognizing the vehicle consists of two stages. First, labeling of two classes (tuktuk and motorcycle), the proposed RCNN modified model is used in the second stage as a classification model that adds some layers and changes the number of filters to categorize two types of vehicles. Based on the created dataset named MAjN_IRAQ dataset, which contains 3017 images of several types of vehicles uploaded to the Git-hub web page, the proposed results model compared with other models as the recognition recall performance metrics, precision, and F1 score.

In this article, the research was organized as follows: Section 1 depicts background information for the research about vehicle detectors and classification. A proposed model for the vehicle recognition system in the streets of Baghdad. A group of layers was adopted for use in conducting the training process on targets that were manually labeled in image plane. The training layers were improved by controlling the values of the number of filters and the filter size in the convolutional neural networks to be more effective and consistent in obtaining on the best features to recognized targets using the RCNN technique. There are two stages in the proposed model architecture for vehicle detection presented in Section 2: classification of two classes and vehicle detection. Section 3 contains a demonstration of the experimental results and discussion, as well as a comparison of classification performance with further techniques utilizing various algorithms. In section 4, the major contributions and conclusions of this study are discussed.

2. METHODOLOGY: VEHICLES CLASSIFICATION RCNN SYSTEM

This study introduced a system that used RCNN with modified layers to recognize two classes of vehicles (Tuktuk and Motorcycle). Here is the most well-known information about the RCNN method: RCNN (Region-based-Convolutional Neural Network) is a popular object detection algorithm that relates both region proposals and deep convolutional neural networks. The RCNN framework consists of three key steps. Firstly, it generates region proposals using algorithms such as Selective Search, which identifies different object regions within an image. Secondly, these region proposals are aligned and warped to a fixed size and shape and then passed through a pre-trained CNN to extract relevant features. Finally, these features are used for object classification and localization using SoftMax classifiers.

To verify the efficacy of the proposed RCNN modified model, contain a new convolutional layer as well as changing the filter size and number in each layer to achieve the features down sampling to get fully connected layer as shown in Table 1. It is implemented to detect multiple vehicles within the dataset created by camera system type Go Pro Hero 9 named MAjN_IRAQ dataset, which is accessible https://github.com/mahaaziz23/MAjN_IRAQ from Baghdad Street of Iraq. In Figure 1, an example MAjN image is shown. The following is a summary of the MAjN dataset specification:

- (1) Six raw RGB videos.
- (2) 3017 obtained and 613 characterized full HD images with a resolution of $2.7k \times 1080$ pixels.
- (3) Several moving objects that are connected to traffic control: humans, motorcycles, cars, taxis, tuktuk, minibuses, and trucks. Among them, tuktuk and motorcycles are used for detection in this article.

Table 1. The architecture of the proposed RCNN-modified model layers

Layer	Kernel Size, Stride	Options	Description
Input image	64 64,3	Training Options	sgdm
Convolution 1	256,1	InitialLearnRate	0.0001
MaxPooling	2*2,2	Verbose	true
RELU	-	Minibatch Size	32
Convolution 2	128,1	MaxEpochs	10,30,50,80,100,150
RELU	-	Shuffle	never
MaxPooling	2*2,2	Verbose Frequency	20
Convolution 3	64,1	Checkpoint Path	tempdir
RELU	-		
MaxPooling	2*2,2		
Convolution 4	32,1		
RELU	-		
Max.Pooling	2*2,2		
Fully Connected	64		
RELU	-		
Fully Connected	3		
SoftMax	-		
Classification	-		

Table 2. The training time, accuracy and loss for proposed RCNN modified model at different Epoch

Epoch	Max. Iteration	Time of Training (sec)	Accuracy at Max Iteration	Loss at Max Iteration
10	7890	1300.24	96.88	0.1222
30	23670	3223.96	96.88	0.0286
50	39450	5170.30	100.00	0.0080
80	63120	8277.60	96.88	0.0283
100	78900	10325.85	100.00	0.0159
150	118350	15301.77	100.00	0.0015



Figure 1. A sample MAjN image in the MAjN dataset

The proposed RCNN model consists of two stages: vehicle labeling and classification. This proposal used the RCNN algorithm modified by changing layers and filter numbers. Table 1 shows the proposed RCNN-modified architecture.

The two classes (tuktuk and motorcycle) were trained using the proposed RCNN-modified model. The training process took place in different Epochs. Table 2 shows the time of accuracy, training, and loss for the proposed RCNN-modified model at different Epochs.

Also, using pre-trained networks, fine-tuning transfer learning, as Alex Net [18], (VGG16 and VGG19) [19], all of them are training at 80 Epoch, and compare them with proposed RCNN modified model.

3. RESULTS AND DISCUSSION

This work's outcome section includes four stages to accomplish the planned RCNN modified model. The first stage collects the videos using a camera system in Baghdad Street, splitting videos into frames and labeling them to create the dataset named MAjN_IRAQ dataset. The second stage explains the training data of different Epochs for two classes (Motorcycle, Tuktuk). In this algorithm, some layers are created, and classification parameters are controlled. The Epoch values are changed according to the following (10, 30, 50, 80, 100, and 150), getting different models and studying the effect of Epoch number on model performance as shown in Table 3. Statistical evaluation parameters were calculated to assess the performance of each of the proposed RCNN-modified models. Where [True Positive (TP) denotes the cases when A positive class is accurately predicted by the model, False Positive (FP) takes place when the model wrongly

expects, for a negative sample, a positive class. False Negative (FN): happens when the model wrongly expects, for a positive sample, a negative class. It means that the model has failed to identify an example as positive when it should have been positive] [20, 21] calculated. Table 3 illustrates the evaluation parameters of two classes as precision, True positive rate (TPR) The true positive rate is the proportion of positive instances that are correctly classified by the model and F1 score is a measure of the harmonic mean of precision and recall. Commonly used as an evaluation metric in binary and multi-class classification, the F1 score integrates precision and recall into a single metric to gain a better understanding of model performance. From Table 3, it is clear that the best result that can be adopted efficiently is (at Epoch=80); therefore, it can be concluded that the classification parameters are the best at 80 Epoch, and less error to detect and recognized in a number of (613) images for training and (546) images for testing, so it was chosen and used in other methods to compare their performance with the RCNN modified model.

The performances of the suggested model are assessed using three metrics: precision and recall (True positive rate) are presented in Eqs. (1) and (2). The precision indicates the accuracy and quality of the model's categorization as assessed by the classification outcomes.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$TPR (True\ positive\ rate) = Recall = \frac{TP}{TP + FN} \quad (2)$$

Table 3. Classification accuracy metrics values for proposed RCNN modified model at various Epoch

Epoch	Classes	P(Precision)	TPR (True Positive Rate)	F1 Score
10	Motorcycle	0.8607	0.8947	0.8773
	Tuktuk	0.9207	0.9117	0.9162
30	Motorcycle	0.9091	0.9091	0.9091
	Tuktuk	0.9100	0.8750	0.8922
50	Motorcycle	0.9220	0.9220	0.9220
	Tuktuk	0.8928	0.9174	0.9049
80	Motorcycle	0.8876	0.9294	0.9080
	Tuktuk	0.9375	0.9292	0.9333
100	Motorcycle	0.9634	0.8777	0.9185
	Tuktuk	0.8231	0.8768	0.8491
150	Motorcycle	0.7816	0.8717	0.8239
	Tuktuk	0.8593	0.9090	0.8835

Precision, True Positive Rate & F1score For tuktuk rec. by proposed RCNN

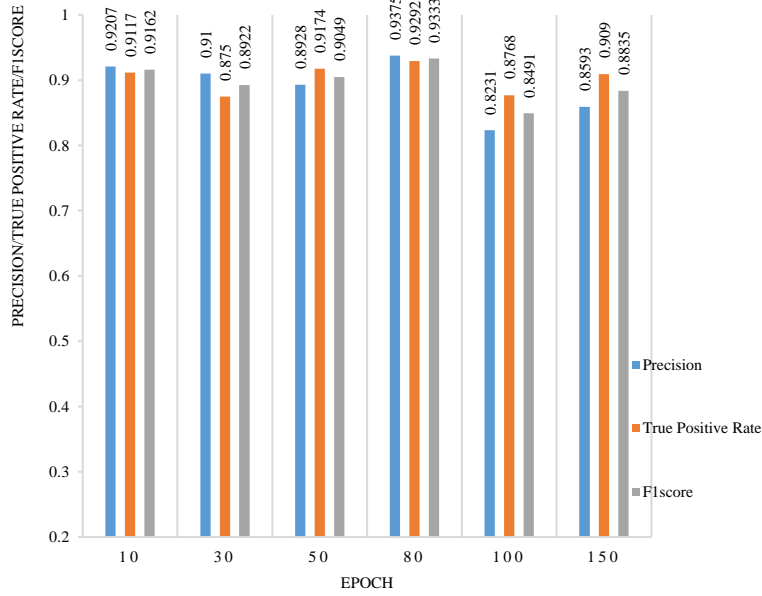


Figure 2. Illustrates the P, TPR, and F! scores for proposed RCNN modified model for the tuktuk class

Precision, True Positive Rate & F1score For motorcycle rec. by proposed RCNN

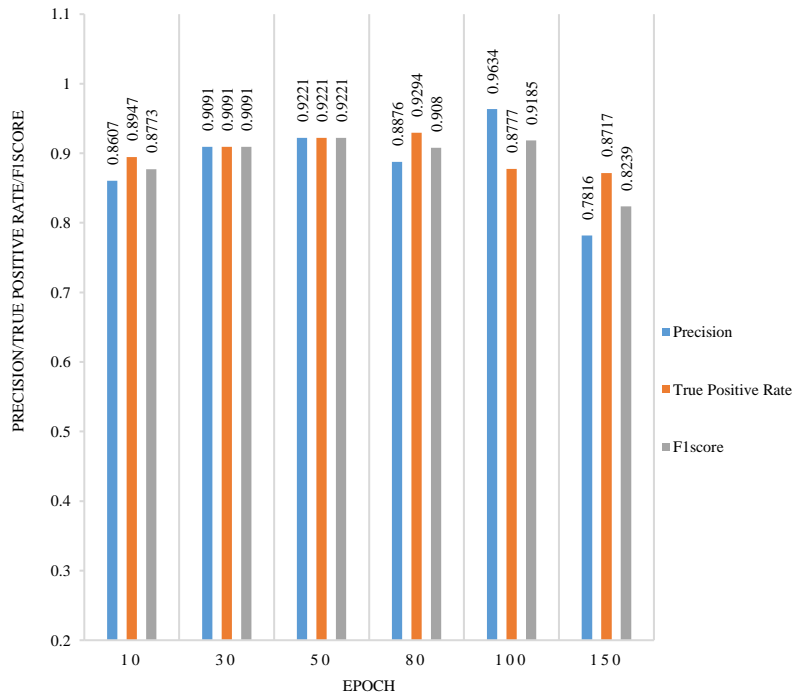


Figure 3. Illustrates the P, TPR, and F! scores for proposed RCNN modified model for the motorcycle class

It's common to refer to the F1 Score as the F Measure or F Score. The quality of the categorization is represented by the F1 score, which demonstrates how well the recall and the precision are balanced, as indicated by Eq. (3).

$$F1 = \left(\frac{2 * (precision * recall)}{(precision + recall)} \right) \quad (3)$$

Figures 2 and 3 depict the classification parameters such as precision, True positive rate TPR, and F1 score for two classes at different Epochs.

Tables 4 and 5 show the effect of the epochs on the accuracy and loss at different iterations.

Figures 4 and 5 illustrate the accuracy and loss of the training for the proposed RCNN-modified model at various iterations.

Table 4. The training accuracy for the proposed RCNN modified model at different iterations

Epoch	Acc. at Max Iteration	Acc. at 0.5 Max Iteration	Acc. at 0.25 Max Iteration	Acc. at 0.1 Max Iteration
10	96.88	93.75	81.25	84.38
30	96.88	100.00	90.63	93.75
50	100.00	100.00	87.50	93.75
80	96.88	93.75	100.00	96.88
100	100.00	100.00	96.88	96.88
150	100.00	100.00	100.00	93.75

Table 5. The training loss for the proposed RCNN modified model at different iterations

Epoch	Loss at Max Iteration	Loss at 0.5 Max Iteration	Loss at 0.25 Max Iteration	Loss at 0.1 Max Iteration
10	0.1222	0.2551	0.6219	0.3106
30	0.0286	0.0091	0.3059	0.2202
50	0.0080	0.0355	0.2481	0.1496
80	0.0283	0.1097	0.0375	0.0709
100	0.0159	0.0059	0.0569	0.1171
150	0.0015	0.0002	0.0185	0.1255

Table 6. The comparison of Alex Net, VGG16, and VGG19 models with proposed RCNN modified model at 80 Epoch

Method	Classes	P(precision)	TPR (True Positive Rate)	F1 Score
Proposed R-CNN	Motorcycle	0.8876	0.9294	0.9080
	Tuktuk	0.9375	0.9292	0.9333
Alex Net.	Motorcycle	0.9512	0.9512	0.9512
	Tuktuk	0.8972	0.8648	0.8807
VGG16	Motorcycle	0.9489	0.9587	0.9537
	Tuktuk	0.875	0.8434	0.8589
VGG19	Motorcycle	0.8217	0.9121	0.8645
	Tuktuk	0.9364	0.8583	0.8966

Table 7. Training and testing time for all models at 80 Epoch

Method	Classes	Testing Time(s)	Average Testing Time(s)	Training Time(s)	Training Time(h)
Proposed R-CNN	Motorcycle	1.87	1.87	8277.60	2.30
	Tuktuk	1.88			
Alex.Net	Motorcycle	1.87	1.89	12658.55	3.52
	Tuktuk	1.90			
VGG16	Motorcycle	98.91	96.85	256331.96	71.20
	Tuktuk	94.79			
VGG19	Motorcycle	18.58	18.56	242786.27	67.44
	Tuktuk	18.55			

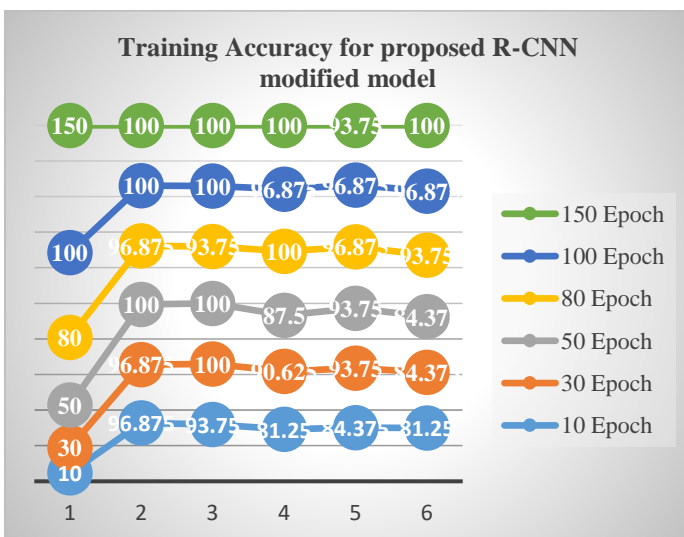


Figure 4. The training accuracy for the proposed RCNN modified model at various iterations

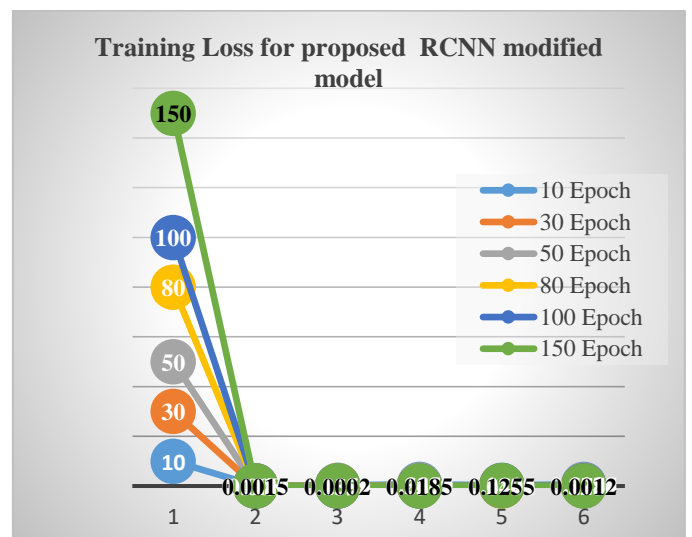


Figure 5. The training loss for the proposed RCNN modified model at various iterations

Precision, True Positive Rate & F1score COMParison For tuktuk AT 80 EPOCH

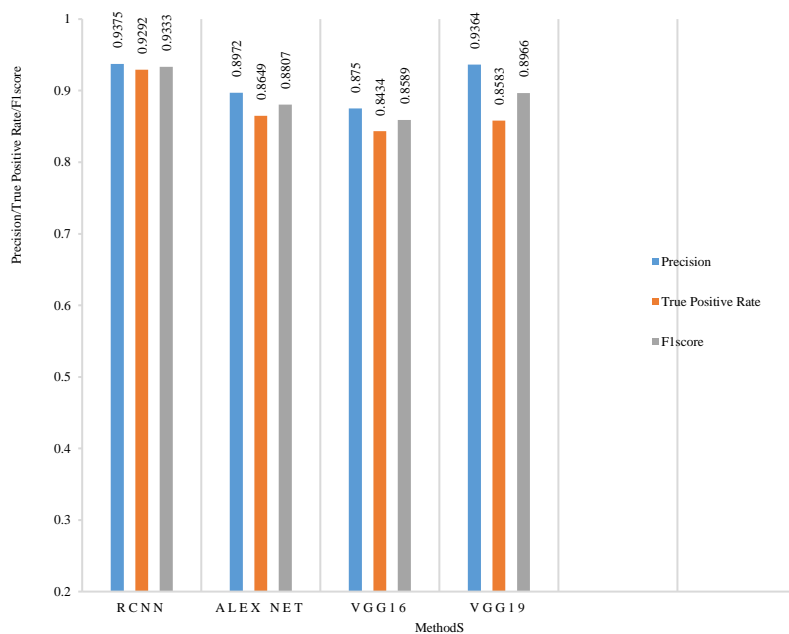


Figure 6. Illustrated P, TPR, F1 score for proposed RCNN modified, Alex Net., VGG16 and VGG19 for tuktuk class

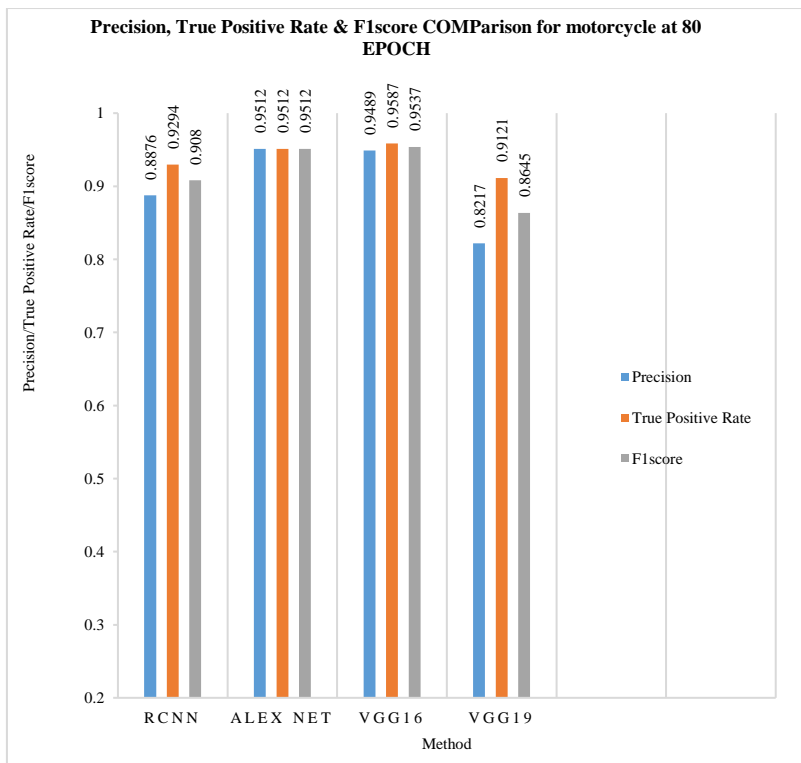
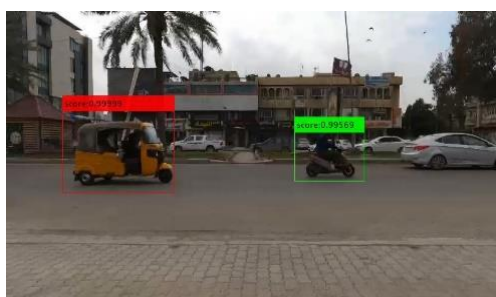


Figure 7. Illustrated P, TPR, F1 score for proposed RCNN modified, Alex Net., VGG16, and VGG19 for motorcycle class



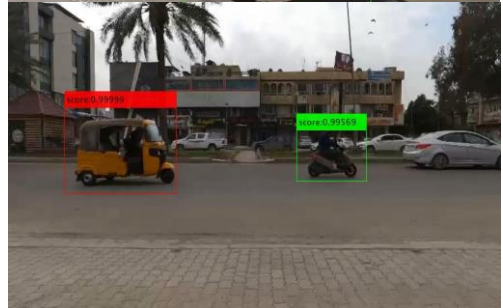




Figure 8. Tuktuk and motorcycle detection by proposed RCNN

The third stage involved training the MAjN_IRAQ dataset using transfer learning with VGG19, VGG16, and Alex Net at 80 epochs, as shown in Table 6. The proposed RCNN modified was compared with the results.

Table 6 illustrates that in the Tuktuk class, the proposed RCNN modified model gets the best results compared with other transfer learning models, while in the Motorcycle class, it gets the best result in the Alex Net, VGG16, and modified RCNN.

Table 7 illustrates the time of testing and training with the proposed RCNN modified model and transfer learning VGG19, VGG16, and Alex Net models.

Table 7 depicts that the testing and training time for the proposed RCNN-modified model for two classes is better than other transfer learning models.

The images of the tuktuk 400 are compared to the images of the motorcycle 300 in training, which led to the appearance of a difference in the results of performance parameters, due to the difficulty of collecting photos from the crowded streets of Baghdad and high temperatures, therefore the RCNN proposed model extracted features of tuktuk more than motorcycle.

This study aims to get the best detection models and choose which one is better.

In the fourth stage, the velocity of the vehicles in the internal streets of Baghdad city ranges between (40-100) km/h. on average, the velocity is equal to 60 km/h (16.66 m/s), and therefore, the distance of 200 m requires a time ($t \cong 12$ s) to pass it. When the velocity equals 100 km/h (27.77m/s) on the highway, the time required to pass the same distance is equal to ($t \cong 7$ s). The conclusion from this is that the proposed RCNN modified, and Alex Net models are working in time.

Figures 6 and 7 illustrate the precision, True positive rate, and F1 score for two classes, Tuktuk and Motorcycle, of the proposed RCNN modified model and compare with other transfer learning models such as VGG16, VGG19, and Alex Net.

Figure 8 shows some sample testing images for two classes by using the proposed RCNN modified model of different images. The results of detected tuktuk and motorcycles marked by red and green boxes in images were classified.

These results can be applied practically in the streets of the capital, Baghdad, in order to allow the vehicles to pass through the streets and does not allow other vehicles to pass on same streets. It is possible to implement the research by using a network consisting of a group of cameras that capture images and send them to the proposed model in order to detect and classify vehicles.

The proposed system was trained on the image in the database MAjN_IRAQ Dataset created in this study can be adopted for detection and distinguishing tuktuk and

motorcycles on the streets of Baghdad with high effectiveness.

4. CONCLUSIONS

In this study, and from the results, it can be concluded:

(1) A camera system was used in the internal streets of Baghdad city to create a dataset, namely MAjN_IRAQ, of various vehicles, including two classes under study.

(2) The proposed RCNN modified model has been used to classify two classes, Tuktuk and Motorcycle, by training the MAjN_IRAQ dataset at various epochs, and the best results were when the Epoch was equal to 80.

(3) The proposed RCNN-modified model increased the performance measures such as precision, recall (True Positive Rate is called TPR), and F1 score.

(4) At the 80 Epoch, the same labeled data was trained using different transfer learning methods such as VGG16, VGG19, and Alex Net. Then, comparing Their models with the proposed RCNN modified model in the classification parameters, it was shown that the Tuktuk class gave the best results while the Motorcycle class gave results whose values were closer to the Alex Net model than to the other models. In addition to the training and testing time with the proposed RCNN modified model, it is less than for the other models.

(5) At 80 Epoch, the precision for the motorcycle class was valued at ($0.8876 \cong 0.89$), while the precision for the Tuktuk class was ($0.9375 \cong 0.94$) for the modified RCNN model. TPR was higher for tuktuk class at ($0.9292 \cong 0.93$), while the lower value was ($0.8434 \cong 0.84$) by using VGG16. The F1 score was higher for motorcycle class with a value of ($0.9537 \cong 0.95$), while the lower value was 0.8589 for the tuktuk class by using VGG16,

(6) The proposed RCNN modified and Alex Net models can be worked on in time.

REFERENCES

- [1] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 580-587. <https://doi.org/10.1109/CVPR.2014.81>
- [2] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In: Cortes C. Lawrence N. Lee D. Sugiyama M. Garnett R. (eds) Advances in Neural Information Processing Systems, Montréal Canada.

- [3] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- [4] Espinosa, J.E., Velastin, S.A., Branch, J.W. (2017). Vehicle detection using Alex Net and faster R-CNN deep learning models: A comparative study. In: Badioze Zaman, H., Robinson, P., Smeaton, A.F., Shih, T.K., Velastin, S., Terutoshi, T., Jaafar, A., Mohamad Ali, N. (eds) *Advances in Visual Informatics*, Cham. https://doi.org/10.1007/978-3-319-70010-6_1
- [5] Joodi, M.A., Saleh, M.H., Kadhim, D.J. (2022). A proposed 3-stage CNN classification model based on augmentation and denoising. *International Journal of Nonlinear Analysis and Applications*, 14(3): 121-140. <https://doi.org/10.22075/ijnaa.2022.27970.3770>
- [6] Arinaldi, A., Pradana, J.A., Gurusinga, A.A. (2018). Detection and classification of vehicles for traffic video analytics. *Procedia Computer Science*, 144: 259-268. <https://doi.org/10.1016/j.procs.2018.10.527>
- [7] Murugan, V., Vijaykumar, V.R., Nidhila, A. (2019). A deep learning RCNN approach for vehicle recognition in traffic surveillance system. In *International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, pp. 0157-0160. <https://doi.org/10.1109/ICCSP.2019.8698018>
- [8] Nguyen, H. (2019). Improving faster R-CNN framework for fast vehicle detection. *Mathematical Problems in Engineering*, 2019: 1-11. <https://doi.org/10.1155/2019/3808064>
- [9] Grents, A., Varkentin, V., Goryaev, N. (2020). Determining vehicle speed based on video using convolutional neural network. *Transportation Research Procedia*, 50: 192-200. <https://doi.org/10.1016/j.trpro.2020.10.024>
- [10] Faruque, M.O., Ghahremannezhad, H., Liu, C. (2019). Vehicle classification in video using deep learning. In *15th International Conference on Machine Learning and Data Mining*, New York, USA, pp. 117-131. <https://www.researchgate.net/publication/346061113>.
- [11] Htet, K.S., Sein, M.M. (2020). Event analysis for vehicle classification using fast RCNN. In *9th Global Conference on Consumer Electronics (GCCE)*, Kobe, Japan, pp. 403-404. <https://doi.org/10.1109/GCCE50665.2020.9291978>
- [12] Algiriyage, N., Prasanna, R., Doyle, E.E., Stock, K., Johnston, D. (2020). Traffic flow estimation based on deep learning for emergency traffic management using CCTV images. In *Proceedings of the 17th International Conference on Information Systems for Crisis Response and Management*, Blacksburg, VA, USA, pp. 100-109.
- [13] Huy, T.N., Duc, B.H. (2020). Traffic flow estimation using deep learning. In *2020 5th International Conference on Green Technology and Sustainable Development (GTSD)*, Ho Chi Minh City, Vietnam, pp. 180-184. <https://doi.org/10.1109/GTSD50082.2020.9303163>
- [14] Sharma, P., Singh, A., Singh, K.K., Dhull, A. (2021). Vehicle identification using modified region based convolution network for intelligent transportation system. *Multimedia Tools and Applications*, 81: 34893-34917. <https://doi.org/10.1007/s11042-020-10366-x>
- [15] Ghasemi Darehnaei, Z., Shokouhifar, M., Yazdanjouei, H., Rastegar Fatemi, S.M.J. (2022). SI-EDTL: Swarm intelligence ensemble deep transfer learning for multiple vehicle detection in UAV images. *Concurrency and Computation: Practice and Experience*, 34(5): e6726. <https://doi.org/10.1002/cpe.6726>
- [16] Djenouri, Y., Belhadi, A., Srivastava, G., Djenouri, D. Lin, J.C.W. (2022). Vehicle detection using improved region convolution neural network for accident prevention in smart roads. *Pattern Recognition Letters*, 158: 42-47. <https://doi.org/10.1016/j.patrec.2022.04.012>
- [17] Alam, M.K., Ahmed, A., Salih, R., Al Asmari, A.F.S., Khan, M.A., Mustafa, N., Mursaleen, M., Islam, S. (2023). Faster RCNN based robust vehicle detection algorithm for identifying and classifying vehicles. *Journal of Real-Time Image Processing*, 20(5): 93. <https://doi.org/10.1007/s11554-023-01344-1>
- [18] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J., Bottou, L., Weinberger, K.Q. (eds) *Advances in Neural Information Processing Systems*, Nevada, USA.
- [19] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, pp. 1-14.
- [20] Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press.
- [21] Chollet, F. (2021). *Deep Learning with Python*. Simon and Schuster.