ZANCO Journal of Pure and Applied Sciences

TYPE Original article
PUBLISHED 05 /02/2024
DOI http://dx.doi.org/10.21271/ZJPAS.36.1.3

OPEN ACCESS

*Corresponding author
Tara Saber Ismail
tara.ismael@su.edu.krd

RECEIVED :23 /11 2022
ACCEPTED :23/02/ 2023
PUBLISHED :05/ 02/ 2024

KEYWORDS:
Deep learning; Yolov5, SSD; Mask R-CNN, DeepSORT, Traffic analysis.

Vehicle Detection, Counting, and Classification System based on Video using Deep learning Models

Tara Saber Ismail*, Abbas M. Ali

Department of Software Engineering and Informatics, College of Engineering, Salahaddin University- Erbil, Kurdistan Region, Iraq

ABSTRACT

Traffic analysis is one of the crucial tasks of intelligent transport system that utilizes deep learning for range of purposes. Many tasks, such as vehicle recognition, vehicle counting, traffic violation monitoring, vehicle speed monitoring, vehicle density and so on, can be accomplished by using cameras installed in strategic locations along roads. In this paper powerful deep learning techniques such as (Yolov5, Mask R-CNN, SSD) and state-of-the-art object tracking algorithm known as DeepSORT was used to perform real time vehicle detection and counting in a video. A new highway vehicle detection dataset with overall of 32,265, instances of four vehicle classes named: bus, car, motorbike, truck was created in this paper and utilized for training vehicle detection and counting system. Result shows that average counting accuracy by using Yolov5 combined DeepSORT reaches to 95% while reaches to 91% by using Mask R-CNN combined DeepSORT and 84% by using SSD combined DeepSORT in hard environment. From the experimental work, counting accuracy by using Yolov5 outperforms other two deep learning techniques.

1. Introduction

Estimating the number of vehicles in a traffic video sequence is a critical component of intelligent transportation systems (ITS), which offers crucial traffic flow information. The number of vehicles on the road shows traffic circumstances such as lane occupancy, congestion level and traffic status which can be utilized for automatic navigation, accident prevention and congestion prevention(Zhang et al., 2011).The main tools for vehicle counting in conventional ITS are specialized sensors like magnetic coil, microwave, or ultrasonic detectors. These sensors have restrictions on the cost of installation and receiving detailed information. Due to the advancement of image processing technology and its potent capabilities, video-based vehicle counting systems have started to gain attention (Liu et al., 2013).These systems provide more traffic parameters, such as detecting vehicle category, density, speed, and traffic accidents for low costs, simple installations, and easy maintenance.

In recent years before deep learning became the standard in computer vision, machine learning and classification techniques were widely utilized. However, the drawbacks of machine learning and classification methods include their high time complexity, weak region selection, and insufficient resilience of manually generated features. As a result, a deep learning approach to target detection is provided(Meng et al., 2020). Deep learning with its large and deep networks, automatically preprocess and extract the image features within its networks then classify the image class, even more it can detect the location of every single object inside the image. However, deep learning demands high machine specifications and a significant amount of data to train networks and maximize their performance (Chauhan and Singh, 2018),(Hagerty et al., 2017). Although deep learning-based target recognition has numerous benefits, it can be challenging to maintain a balance between accuracy and speed. Real-time execution on a constrained computer platform is still a challenging issue(Chen et al., 2018).

In this paper the powerful deep learning techniques was used to perform vehicle counting in Kurdistan region. The main contribution of this article is the creation of a new highway vehicle detection dataset, as well as the analysis and study of Kirkuk traffic roads for maintenance planning.

This paper organized as follows. Section two is material and methods. Section three is result. Section four is discussion. Section five is conclusion.

2. Materials and methods

2.1 Literature review

(Tsai et al., 2018, Haritha et al., 2021, Bautista et al., 2016, Hicham et al., 2018, Ambata et al., 2019, Ullah and Lee, 2017)used a basic CNN architecture for vehicle detection and counting in urban roads. (Tsai et al., 2018, Haritha et al., 2021)modified CNN architecture to be robust to different weather conditions and scale vary limitations. However,(Bautista et al., 2016)used CNN architecture for low resolution images taken from camera, while(Hicham et al., 2018) used CNN architecture to solve the issues of imbalanced vehicle dataset for vehicle recognition.(Ambata et al., 2019) used CNN for solving the issues of congestion and (Ullah and Lee, 2017) for extracting vehicle information. In addition to basic CNN architecture (Li et al., 2021, Sun and Hu, 2021, Al-Ariny et al., 2020) used modified version of CNN such as (Fast R-CNN, Faster R-CNN, Mask R-CNN, Yolo).(Li et al., 2021) used Faster R-CNN to accurately detect vehicles in day and night time and leveraged Faster R-CNN with domain adaptation via transfer learning for night time vehicle detection.(Sun and Hu, 2021) used Fast R-CNN as analysis base of urban roads. In addition to above modified CNN architecture (Al-Ariny et al., 2020)used improved Mask R-CNN for dealing the issues of congestion and occlusion.

(Lou et al., 2019, Cepni et al., 2020, Sudha and Priyadarshini, 2020, Ligayo et al., 2021, Zuraimi and Zaman, 2021) used yolo-v3 as a novel object detection technique for vehicle detection. (Gupta et al., 2022)used tiny-yolov3 for military vehicle detection and classification in real-time environment, while (Song et al., 2019)used yolo-v3 to solve the issues of small object detection and multi-scale variation of the object in highway management, and (Rashmi and Shantala, 2020)used for analyzing vehicle density in urban roads and (Jin et al., 2021)to analyze urban traffic data in urban logistics. In addition (Azimjonov and Özmen, 2021)improved yolo model by changing the classifier and replaced with machine learning classifier, and (Taheri Tajar et al., 2021)enhanced a yolo model by pruning and simplifying some of the unnecessary layers to be used as a lightweight vehicle detection and classification system for low power systems, while (Sudha and Priyadarshini, 2020) enhanced yolov3 model to be more robust to weather conditions.

(Meng et al., 2020) (Chen et al., 2018, Harikrishnan et al., 2021) used SSD as base of vehicle detection.(Meng et al., 2020) used to solve the issues of occlusion and multi tracking for different driving directions. However, (Chen et al., 2018) modified SSD to fast SSD (Chen et al., 2018) improved limited computations and occlusion issues.(Harikrishnan et al., 2021) replaced original SSD with Inception-SSD to solve the issues of multi-scale vehicle detection in various weather and traffic conditions. To analyze vehicle density in the urban roads (Ham et al., 2020) used various deep learning techniques (faster R-CNN), R-FCN, SSD). (Kausar et al., 2020)focused on the problem of two-wheel object detection in extreme climate condition by using one stage detector (SSD, SDDLite, YOLOv3) and two stage detectors (RCNN, Fast-RCNN, Faster-RCNN, R-FCN).

2.2 Methodology

This section outlines the techniques that was used for study on vehicle detection and counting in Kurdistan region. It describes the dataset creation up to testing. This section also explains how the robust state-of-the-art object detection techniques such as (Yolov5, Mask R-CNN, SSD) combined with the efficient online tracking DeepSORT and utilized counting function based on polygon zone to perform vehicle detection and counting in highway.

2.2.1 Data Collection and Description

The dataset was collected from various sources such as IStock videos form google, open-source dataset by (Song et al., 2019), and manually recorded videos of highways in Kurdistan Region. (Song et al., 2019)The dataset image is from a Hangzhou, China, captured from 23 surveillance cameras which contain vehicles with dramatic scale variation. The high-definition dataset (Song et al., 2019)has overall of 11,129 images, but only 250 images form high-definition dataset was used as a first part of data collection. Istock videos has a total of 406,623, videos only 320 traffic videos of (1-2) minute duration of highway was used as a second part of data collection. Lastly, the third type of data was collected form manually recorded videos of various places in Erbil city such as 150-meter road and Kirkuk road and Koya road under different weather conditions such as sunny and cloudy and rainy. The dataset consists of four classes including bus, car, motorbike, truck and labeled and annotated by Roboflow annotation tool then exported into yolo txt file format to train yolov5 model and pascal voc xml file format to train Mask-RCNN, SSD model. The created new highway vehicle detection consists of 3786 image frames, and has 32,265 instances overall. Figure 1shows sample of the dataset. Table 1 shows information about the new highway dataset.

Table 1:Dataset information

	Images	Class	Instances
Train 70%	2650	bus	2,026
Val 20%	757	car	23,499
Test 10%	379	motorbike	1,796
		truck	4,944
Total	3786	32,265 instances	

Figure 1. Sample of dataset

2.2.2 System Architecture

Even though the most majority of real time vehicle detection and counting system have already been put in place, new developments in AI, object detection, and tracking are always being made, bringing new capabilities and making substantial advancements over one another. The performed study on vehicle detection and counting used state-of-the-art object detection techniques such as (Yolov5, Mask-RCNN, SSD) and combined with efficient tracker named DeepSORT to track detected vehicles and assign IDs to them then used counting method to count four vehicle classes as they passed through polygon zone. Overall block diagram of system is shown in figure 2.

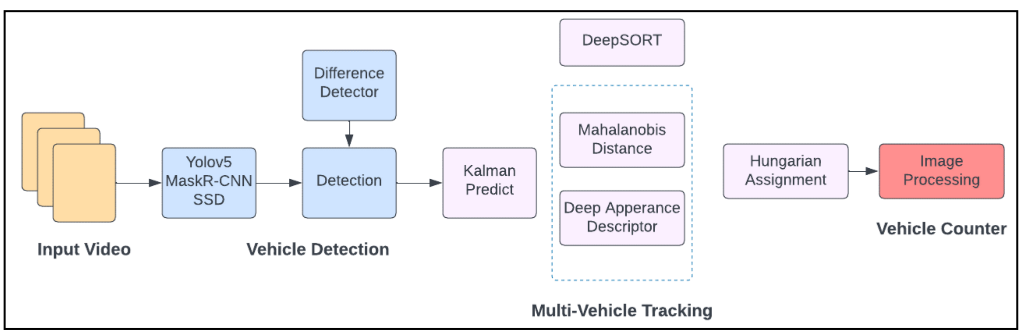

Figure 2. Overall block diagram of system

2.2.3 Hardware Requirements

The field of computer vision has advanced significantly in recent years due to the development of deep learning theory and GPU hardware devices. It has significant practical implications to reduce the amount of manpower by using computer vision technologies. MSI machine of processor 11th Gen Intel(R) Core (TM) i7-11800H @ 2.30GHz 2.30 GHz and RAM 16 GB and NIVIDIA GeForce RTX3070 was used for implementing and testing the study on vehicle detection and counting system.

2.2.4 Deep learning models used in this work

Object detection is a fundamental component of computer vision and digital image processing, as well as the basis of intelligent monitoring systems used in a variety of application use cases. In this study vehicle detection based on leveraging transfer learning procedure of efficient deep learning techniques (Yolov5, Mask R-CNN, SSD).

1.YoloV5

Yolo object detection method is the single-stage object detection method proposed by(Redmon and Farhadi, 2018). It unifies classification and bounding box into a regression problem and eliminates the stage of candidate box extraction in two-stage method. The YOLO algorithm works in the manner described below: initially the image is split into S × S meshes. Each grid is responsible for determining the target where the actual box will fall the middle of the grid. An overall of S × S × B bounding boxes are produced from these meshes. Every bounding box has five parameters includes target width and height dimension and target center point coordinate and confidence score of containing an object(x,y,w,h,c). S × S grids predict the target's category possibility in that grid. The category score for each prediction box is then calculated by multiplying the category probability by the prediction bounding box confidence. The final prediction results are obtained by filtering these prediction boxes using non-maximum suppression (NMS). The YOLO series' algorithms have advanced quickly in recent years(Redmon and Farhadi, 2018). Two variations of YOLO v4 and v5 successively released in 2020. The YOLO v5 was made available by Glenn Jocher in 2020. The input, backbone, neck, and prediction of the network are all displayed in figure 3. YOLOv5 presents its model in five various scale forms (nano, small, medium, large, extra-large). The only difference between each of these scales is the model's extended depth and width; as a result, the model's overall structure is unchanged, but it's complexity and size increase. Additional neurons, more hidden layers, batch normalization, and weight initialization can all be used to change the sample architecture.

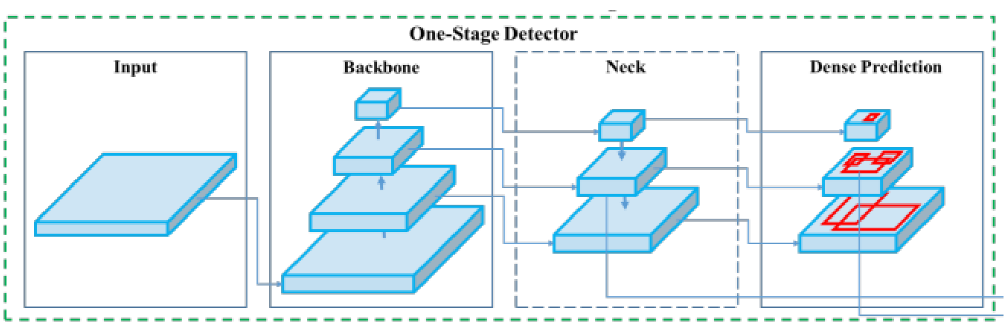

Figure 3.Yolo architecture

2.Mask R-CNN

Mask R-CNN is a theoretically simple, adaptable, and universal framework for object recognition, detection, and instance segmentation. It is capable of accurately detecting objects in an image and producing a high-quality segmentation mask for each instance. The initial block structure of Mask R-CNN, feature pyramid networks (FPNs), is in charge of feature extraction for object detection(Lin et al., 2017). The second component of Mask R-CNN, the regional proposal network (RPN) (Fattal et al., 2017), collaborates with the detection network and utilizes full-image convolutional features to provide practically cost-free area proposals(Uijlings et al., 2013). Then, by adding a branch for object mask prediction in addition to the one for bounding box identification already there, Faster R-CNN was further developed to create the Mask R-CNN. Mask R-CNN as shown in figure 4 adds mask branches to obtain high precision instance segmentation from pixel-to-pixel alignment in order to be able to implement the mask function. Target recognition, detection, and segmentation are three tasks that Mask R-CNN is capable of completing. It still has a 5 FPS detection speed.

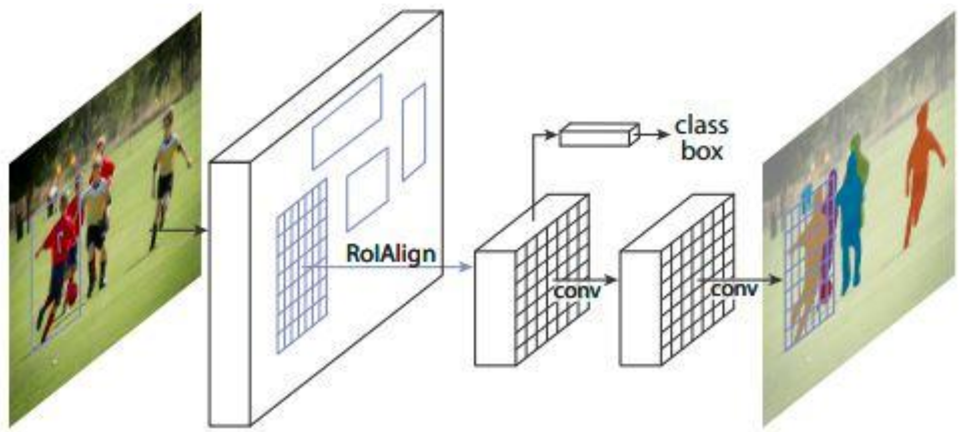

Figure 4. Mask-RCNN architecture

3.Single Shot MultiBox Detector (SSD)

One of the most popular deep learning-based object recognition frameworks at the moment is SSD, or "single shot multibox detector." Wei Liu first brought out SSD at the 14th European Conference on Computer Vision (ECCV) in 2016, and it has since developed into another one-stage object recognition approach that has drawn a lot of interest following YOLO(Liu et al., 2016). In addition to borrowing from the Faster R-CNN anchor's mechanism and feature pyramid structure, SSD also adopts YOLO's regression concept and successfully implements the detection and classification of multiple bounding boxes based on a straightforward end-to-end network. SSD has a faster detection time as compared to Faster R-CNN because it does not need candidate region extraction. When compared to YOLO, SSD uses a fully-connected layer less frequently and has better detection accuracy.

Figure 5 shows the architecture of SSD. Three components make up the SSD network model: the basic network, the feature extraction network, and the detection network. Based on VGG16, the fundamental network is enhanced (visual geometry group 16). The last two fully-connected layers, FC6 and FC7, are exchanged out for convolutional layers Conv6 and Conv7 because they will interfere with the position information of the features if they are fully-connected. Then, Conv8, Conv9, Conv10, and Conv11 are inserted as the final four sets of convolutional layers. Each layer uses 3× 3 convolutional kernels for feature extraction and 1 ×1 convolutional kernels for dimension reduction. Next, a multi-scale feature extraction network in the shape of feature pyramids is created by combining the feature maps of Conv4_3 and Conv7 with those of Conv8_2, Conv9_2, Conv10_2, and Conv11_2. Finally, each feature map in the detection network is subjected to convolutional operations using two convolutional kernels of size 3×3. Category confidences are output by one convolutional kernel, while object position data is provided by the other for regression. The combined calculation results are sent to the loss layer. The non-maximum suppression (NMS) algorithm is used to provide the final detection result (Liang et al., 2018).

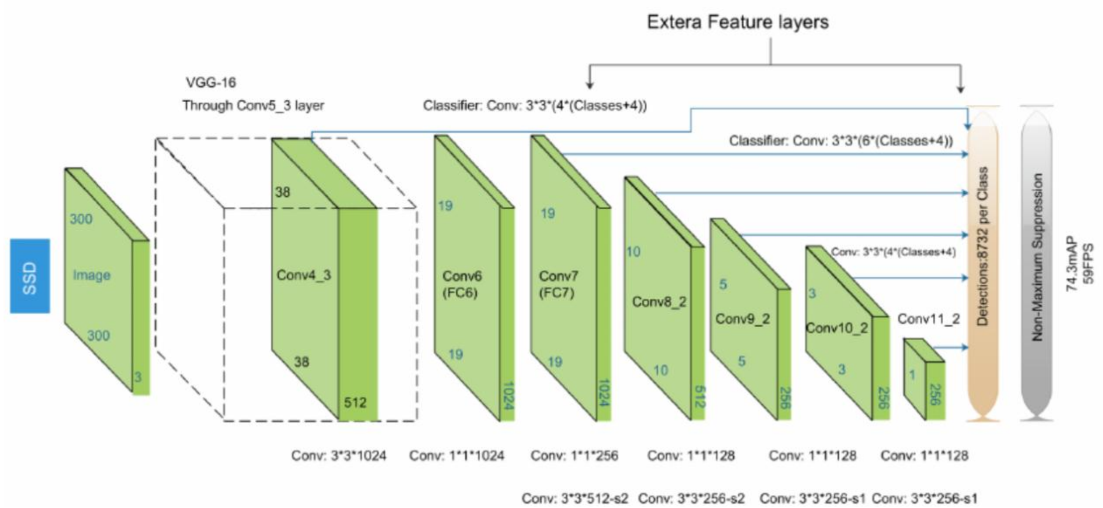

Figure 5. Architecture of Single Shot MultiBox Detector (SSD)

2.2.5 Vehicle Tracking

The DeepSORT multi-object tracking technique is then used to match the extracted vehicle features with the other video frames in order to achieve a correlation between the same vehicle and other comparable vehicles. This is done after the vehicles have been recognized by using one of the deep learning detection techniques (Yolov5, Mask R-CNN, SSD). For tracking, the DeepSORT algorithm combines the Kalman Filter and Hungarian algorithm. The Kalman Filter is used to estimate the current state of a vehicle based on some prior value and to provide the associated uncertainties (Gunjal et al., 2018); the Mahalanobis distance is then used to include the Kalman Filter's predicted uncertainties. After the vehicle location has been determined, the Hungarian algorithm is then used for the vehicle association and ID attribution, assigning a unique identity to the vehicle and determining whether the vehicle seen in the current frame is the same as that seen in the previous frame. The flow chart of DeepSORT tracking algorithm is shown in figure 6.

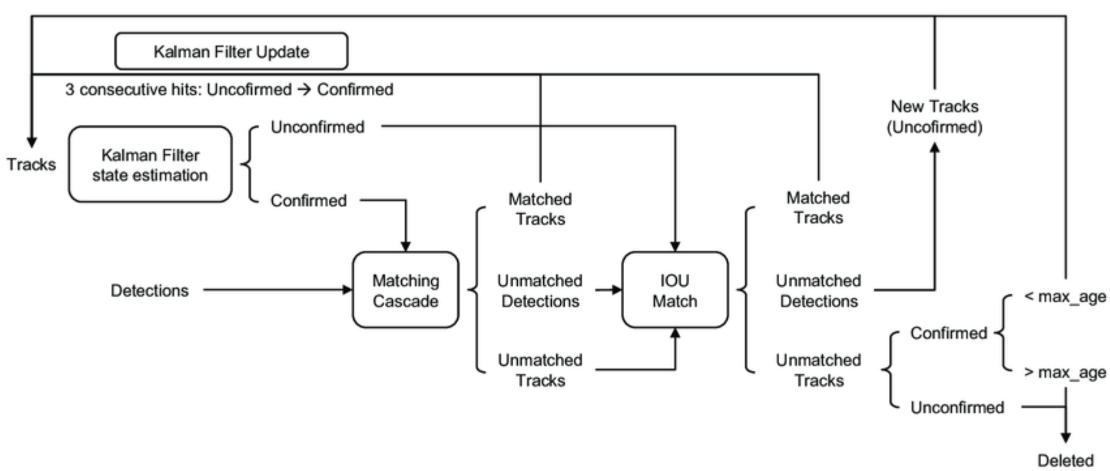

Figure 6: DeepSORT tracking algorithm flowchart

2.2.6 Vehicle Counting

Since some vehicles have extremely similar features, ID changes may occasionally happen during tracking, leading to misleading vehicle counts. Therefore, "virtual polygon area" was added for precisely counting the total number of vehicles that have crossed the highway in order to strengthen the robustness of the vehicle counter by not relying simply on the vehicle tracker and ID assignment for the counting. Two virtual polygon area was drawn one in the right direction and other in the left direction of the Kirkuk Road experiment. The counting function was set to count only the vehicles whose center coordinate passed through the polygon region and their tracking ID remained unique.

3. Results

3.1 Training results

The deep learning techniques (Yolov5, Mask R-CNN, SSD) was trained on 2650 training images and was evaluated on 757 validation images of the created highway vehicle detection dataset to perform real-time vehicle detection and classification in the video. The main metrics that was used to evaluate the trained models are the loss plot function and mean average precision mAP@.5.

As a result of excessive data and heavy computation training time required for deep learning models, transfer learning and pretrained weights was utilized which have been originally trained on Ms COCO dataset as a starting point for models training. The first model was trained by using (YOLOv5m6) version of Yolov5 with random data augmentation of translation 0.1, scaling 0.5 and setting hue-saturation value to (0.015, 0.7, 0.4) and resizing input image to (640 * 640) pixels and set the initial learning rate to (0.01) with momentum to (0.937) and weight decay to (0.0005). To improve the model performance, stochastic gradient descent (SGD) was employed as an optimizer. The yolov5 model trained for (100) epochs which took nearly 2 hours.

The second model was trained by using Mask R-CNN. To reduce training time, only the head layer of the network backbone was trained and image resizing mode was set to square and image min dimension was set to 800px and image max dimension was set to 1024px with steps per epoch to 500. Stochastic gradient descent was utilized as an optimizer for network with learning rate set to 0.001, and weight decay set to 0.0001, and learning momentum set to 0.9. The model was trained successfully which took nearly 16 hours to complete.

The last model was trained by using single shot multibox object detector (SSD) with the super performance optimizer stochastic gradient. If the learning rate is set to big then the model is not converging, oppositely if is set too small then the speed of network model will be slow down so the initial learning rate was set to 1e-3, weight decay was set to 5e-4, and momentum weight was set to 0.9. And to prevent overlearning the features of the training set and the possibility of over-fitting, the L2 regularization for the loss function was used. In this study the total number of epochs was set to 100 which took around 7.5 hours to finish successful training.

To evaluate, the performance of the test, the precision/recall curve, including precision (P), recall (R), and mAP (PR-curve) were measured and compared. From the model's confidence threshold, the PR-curve was calculated. The recall is the percentage of all positive samples found above the 50% confidence level, and the precision is the percentage of all positive samples found at the same level of confidence. P and R calculated as presented in equation 1 and 2

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

$$\text{Recall (R)} = \frac{TP}{TP+FN} \quad 2$$

where TP, FP, and FN stand for the number of true positives, false positives, and false negatives, respectively. The greatest precision measured for a model whose corresponding recall surpasses the recall level is used to interpolate the precision at each recall level in the PR-curve (r). Figure 10 shows precision and recall curve of trained Yolov5 in which class car has highest precision and recall and class motorbike has the lowest precision and recall. Fig 11 shows precision and recall of trained SSD in which class bus has the highest precision and recall and class motorbike has the lowest precision and recall. In both figure 10 and 11 the Yolov5 and SSD models perform well due to the models are near to ideal model.

The mAP specifies the mean of the average precision (AP) of the whole number of classes (n), which in our example is 4. The average precision on a set of 11 equally spaced recall levels [0, 0.1,..., 1] is called as the AP, which summarizes the PR-curve.

$$mAP = \frac{\sum AP}{n} \quad 3$$

Table 2 shows average precision (AP) of trained Yolov5 and SSD model. The trained Yolov5 model obtained highest average precision of 0.889 by car class while the lowest average precision of 0. 617 by motorbike class. However, the trained SSD model obtained highest average precision of 0.9068 by bus class while the lowest average precision of 0.6297 by motorbike class. The overall highest mean average precision (mAP@.5) is 0.832 achieved by SSD model. Table 3 shows overall mean average precision(mAP@.5) of trained models.

Table 2:Average precision of vehicle classes of trained models.

Model	Bus	Car	Motorbike	Truck
Yolov5	0.859	0.889	0.617	0.783
SSD	0.9068	0.8934	0.6297	0.8985

Table 3:Overall Mean average precision of Yolov5, Mask R-CNN, SSD

	Yolov5	Mask R-CNN	SSD
Overall mAP@.5	0.787	0.765	0.8321

3.2 Experimental results

Real time vehicle detection and counting in Kirkuk Road located in Erbil city, Kurdistan region was performed in this study by situated camera at the pedestrian bridge near college of engineering as shown in figure 12. Several videos of different traffic conditions were tested to perform evaluation. In this study a busy traffic video of duration (00:21) seconds was used as input to the system to perform real-time inferences on various models. Yolov5 achieved counting accuracy of 90% at right side of the road while achieved 99% of accuracy at left side of the road. However, SSD achieved 78% of accuracy at right side of the road, while 90% of accuracy at left side of the road. Lastly Mask R-CNN achieved counting accuracy of 94% at right side of the road, while 87% of accuracy at left side of the road. Table 4 shows counting result of right side of the road. Table 5 shows counting result of left side of the road. Finally average counting accuracy of used (Yolov5, SSD, Mask R-CNN) techniques are (95%,84%,91%) respectively. Table 6 shows average counting results of experiment.

Table 4. Counting Result of Yolov5, Mask R-CNN, SSD of right side

Video name	*Model*	*Vehicle*	*Counting right side*		*Counting error*	*Accuracy*
			Real	*System*		
Video_4453 00:00:21 Right side of road	Yolov5	Bus Car Motorbike Truck	2 74 1 4	0 76 0 8	+2 +2 +1 +4	90%
Video_4453 00:00:21 Right side of road	SSD	Bus Car Motorbike Truck	2 74 1 4	0 62 0 8	+2 +12 +1 +4	78%
Video_4453 00:00:21 Right side of road	Mask R-CNN	Bus Car Motorbike Truck	2 74 1 4	1 72 0 3	+1 +2 +1 +1	94%

Table 5:Counting Result of Yolov5, Mask R-CNN, SSD of left side

Video name	*Model*	*Vehicle*	*Counting left side*		*Counting error*	*Accuracy*
			Real	*System*		
Video_4453 00:00:21 Left side of road	Yolov5	Bus Car Motorbike Truck	1 51 3 2	1 51 3 3	+0 +0 +0 +1	99%
Video_4453 00:00:21 Left side of road	SSD	Bus Car Motorbike Truck	1 51 3 2	0 48 2 3	+1 +3 +1 +1	90%
Video_4453 00:00:21 Left side of road	Mask R-CNN	Bus Car Motorbike Truck	1 51 3 2	0 56 1 1	+1 +5 +2 +1	87%

Table 6:Average counting result.

Video name	*Model*	*Average Accuracy*
Video_4453 00:00:21	Yolov5	95%
Video_4453 00:00:21	SSD	84%
Video_4453 00:00:21	Mask R-CNN	91 %

4. Discussion

The main metrics that was used to evaluate trained models are the loss plot function and mean average precision mAP@.5 as shown in figure 7,8,9 and table 3.

Mean average precision(mAP@.5) is the reliable metric used to evaluate object detection models such as Yolo, Mask R-CNN, SSD. Experimental Result of all three trained models (Yolov5, Mask R-CNN, SSD) shows that SSD achieved highest mean average precision of 0.8321, while Mask R-CNN achieved lowest mean average precision of 0.765. Class car obtained highest mean average precision in all three trained models, while class motorbike achieved lowest mean average precision in all three trained models due to the class of car has more than 20,000 instances while motorbike has less than 2000 instances and leads to misclassification.

The loss plot function was used as a second metric to evaluate the model's performance of three detection techniques (Yolov5, SSD, Mask R-CNN) that were trained on 100 epochs. The loss plot function of trained Yolov5 shows that validation class loss decreased from 0.023809 to 0.0067695 as shown in figure 7, while loss plot function of trained SSD shows that validation class loss decreased from 3.2599 to 1.2912 as shown in figure 8. However, loss plot function of trained Mask R-CNN shows that validation class loss decreased from 0.0709 to 0.0471 as shown in figure 9. The loss plot function of three trained models Yolov5, SSD, Mask R-CNN demonstrates that yolov5 converges faster than other two models.

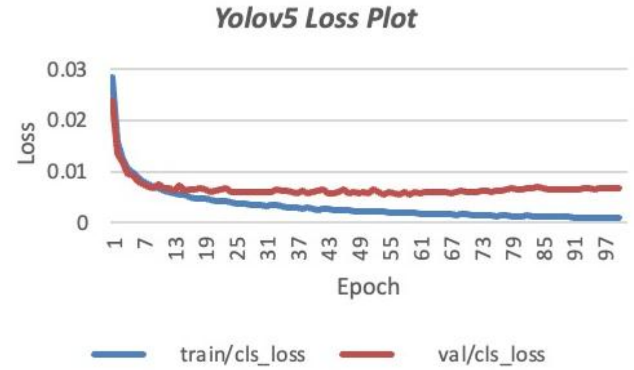

Figure 7: Loss plot function of trained Yolov5 over 100 epochs

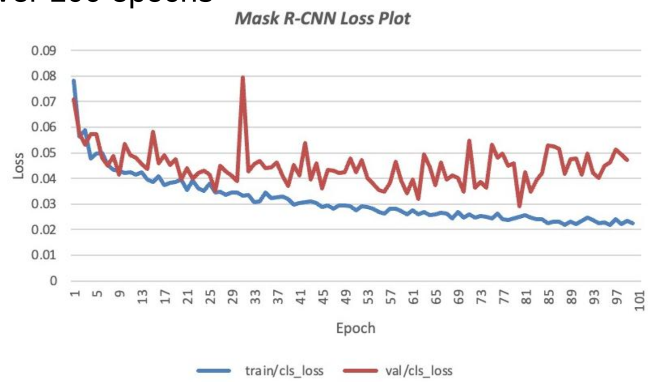

Figure 8: Loss plot function of trained Mask R-CNN over 100 epochs

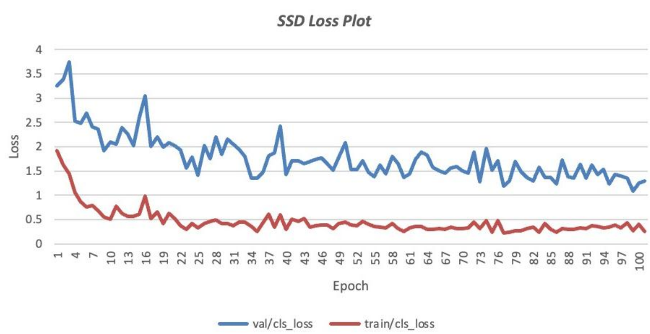

Figure 9: Loss plot function of trained SSD-512 over 100 epochs

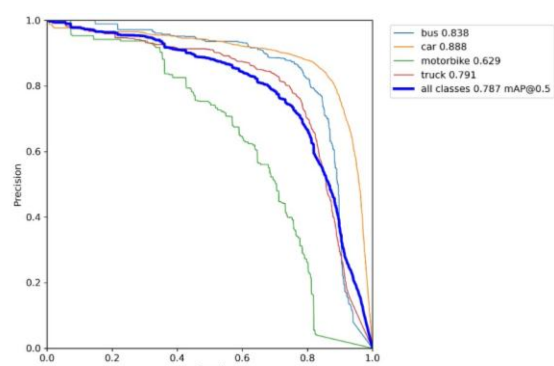

Figure 10: precision and recall curve of trained Yolov5

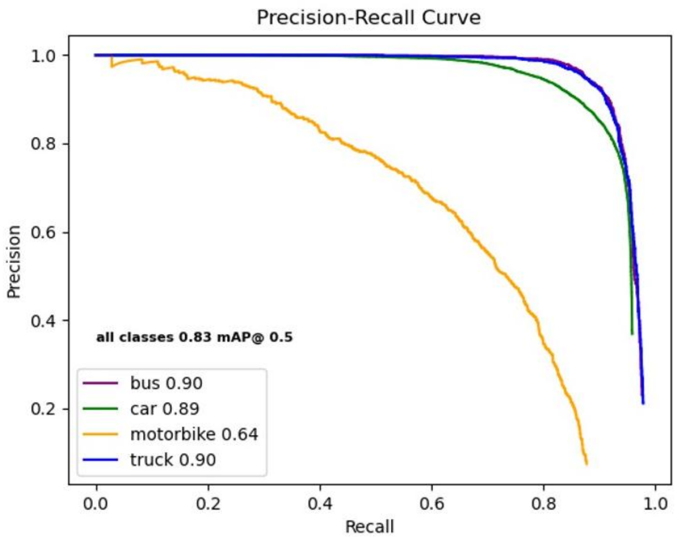

Figure 11: precision and recall curve of trained SSD

With regard to vehicle counting the experimental result shows that by using Yolov5 average counting accuracy achieved is 95% while by using Mask R-CNN average counting accuracy achieved is 91% and by using SSD average counting accuracy achieved is 84%. Figure 13 shows the average counting accuracy of experimental test. Experiments result demonstrates that Yolov5 outperforms other two models and all three trained models achieved better counting results at the left side of the road and has a good capability to scaling and occlusion. The performed test on vehicle counting compared to several methods such as (Yang and Qu, 2018) which used background subtraction and Kalman filter to track detected vehicles in a video and achieved average counting accuracy of 92.2 %. (Lou et al., 2019) which used yolov3 with modified Kalman filter obtained average counting accuracy of 92.1 %. (Bhaskar and Yong, 2014) which used GMM and Blob detection and achieved average counting accuracy of 91%.(Ambata et al., 2019) used Faster-RCNN and centroid tracker algorithm achieved average counting accuracy of 71.65%. (Rashmi and Shantala, 2020) used blob analysis and yolov3 for vehicle detection and counting in a video obtained average counting accuracy of 68.5% and 89.3% respectively. Also (Song et al., 2019) used Yolov3 and OBR detection technique and used virtual line to count the car, bus, motorbike, truck obtained average counting accuracy of 93.2% while (Fachrie, 2020) used yolov3 and virtual line counter to count car, bus ,truck near to our test achieved average counting accuracy of 92.20%. However, our trained system tested in busy traffic and faced several issues such as severe brightness, occlusions, small scale vehicles, but it achieved average counting accuracy of 95% by using yolov5 which outperforms above researchers. Table 7 shows the multi vehicle counting algorithm performance evaluation with other existing works.

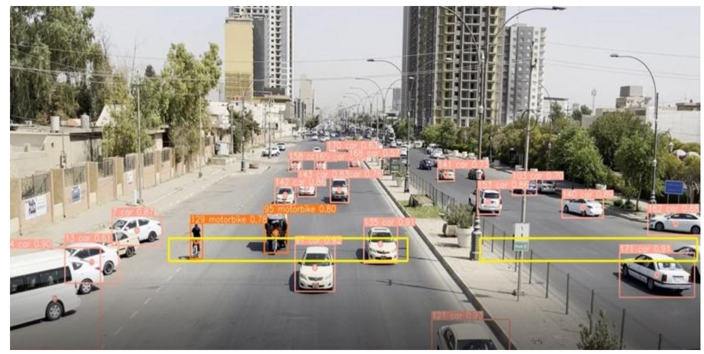

Figure 12: Experiment of Kirkuk Road

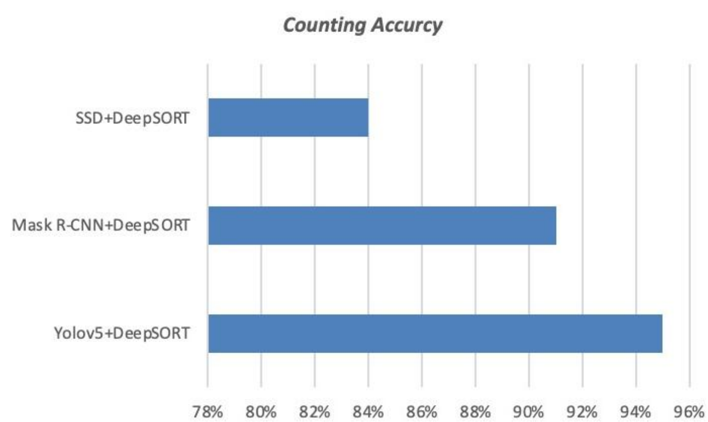

Figure 13: Average counting accuracy of experiment.

Table 7: Multi vehicle counting algorithm performance evaluation with other existing works.

Method	***Detector***	***Tracker***	***Counting Accuracy***
(Yang and Qu, 2018)	Background Subtraction	Kalman Filter	92.2 %
(Lou et al., 2019)	Yolov3	Modified Kalman Filter	92.1 %.
(Bhaskar and Yong, 2014)	GMM and Blob detection	No Tracker	91%
(Ambata et al., 2019)	Faster R-CNN	Centroid Tracking	71.65%

(Rashmi and Shantala, 2020)	blob analysis and Yolov3	No Tracker	89.3%
(Song et al., 2019)	Yolov3	OBR	93.2%
(Fachrie, 2020)	Yolov3	No Tracker	92.20%
Ours	Yolov5 Mask R-CNN SSD	DeepSORT	95% 91% 84%

5. Conclusions

In this paper the most powerful deep learning techniques was utilized to perform real time vehicle detection and counting in Kurdistan region. Object detection result shows that SSD outperforms other state-of-the-art object detection techniques that achieved overall mAP@.5 of 83%. Car class achieved highest mean average precision however, motorbike class achieved lowest mean average precision in all state-of-the-art object detection techniques due to the lack of training data and small scale of motorbikes in the dataset. Experimental result demonstrates that Yolov5 combined DeepSORT achieved highest average counting accuracy which is 95 % while SSD combined DeepSORT achieved lowest average counting accuracy which is 84% and Mask R-CNN combined DeepSORT obtained average counting accuracy which is 91%. Counting system based Yolov5 outperforms other counting models in complex and busy traffic environment and due to their performance, it can be used as main object detection model for real time vehicle detection and classification. In future work we will plan to analyze results of performed test by using data mining techniques.

Acknowledgment: Not applicable.

Financial support: No financial support.

Potential conflicts of interest. All authors report no conflicts of interest relevant to this article.

References

Al-Ariny, Z., Abdelwahab, M. A., Fakhry, M. & Hasaneen, E.-S. An efficient vehicle counting method using mask r-cnn. 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), 2020. IEEE, 232-237.

Ambata, L. U., Del Castillo, I. a. P., Jacinto, J. R. H. & Santos, C. M. T. Public and private vehicle quantification and classification using vehicle detection and recognition. 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), 2019. IEEE, 1-6.

Azimjonov, J. & Özmen, A. J. a. E. I. 2021. A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways. 50, 101393.

Bautista, C. M., Dy, C. A., Mañalac, M. I., Orbe, R. A. & Cordel, M. Convolutional neural network for vehicle detection in low resolution traffic videos. 2016 IEEE Region 10 Symposium (TENSYMP), 2016. IEEE, 277-281.

Bhaskar, P. K. & Yong, S.-P. Image processing based vehicle detection and tracking method. 2014 International Conference on Computer and Information Sciences (ICCOINS), 2014. IEEE, 1-5.

Cepni, S., Atik, M. E. & Duran, Z. J. B. J. O. M. C. 2020. Vehicle detection using different deep learning algorithms from image sequence. 8, 347-358.

Chauhan, N. K. & Singh, K. A review on conventional machine learning vs deep learning. 2018 International conference on computing, power and communication technologies (GUCON), 2018. IEEE, 347-352.

Chen, L., Zhang, Z. & Peng, L. J. I. I. T. S. 2018. Fast single shot multibox detector and its application on vehicle counting system. 12, 1406-1413.

Fachrie, M. J. J. R. 2020. A simple vehicle counting system using deep learning with YOLOv3 model. 4, 462-468.

Fattal, A.-K., Karg, M., Scharfenberger, C. & Adamy, J. Saliency-guided region proposal network for CNN based object detection. 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017. IEEE, 1-8.

Gunjal, P. R., Gunjal, B. R., Shinde, H. A., Vanam, S. M. & Aher, S. S. Moving object tracking using kalman filter. 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), 2018. IEEE, 544-547.

Gupta, P., Pareek, B., Singal, G., Rao, D. V. J. M. T. & Applications. 2022. Edge device based military vehicle detection and classification from uav. 81, 19813-19834.

Hagerty, J., Stanley, R. J. & Stoecker, W. V. Medical image processing in the age of deep learning. Proceedings of the 12th international joint conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), 2017. 306-311.

Ham, S. W., Park, H.-C., Kim, E.-J., Kho, S.-Y. & Kim, D.-K. J. T. R. R. 2020. Investigating the influential factors for practical application of multi-class vehicle detection for images from unmanned aerial vehicle using deep learning models. 2674, 553-567.

Harikrishnan, P., Thomas, A., Gopi, V. P., Palanisamy, P. & Wahid, K. a. J. a. I. 2021. Inception single shot multi-box detector with affinity propagation clustering and their application in multi-class vehicle counting. 51, 4714-4729.

Haritha, H., Thangavel, S. K. J. I. J. O. C. & Applications. 2021. A modified deep learning architecture for vehicle detection in traffic monitoring system. 43, 968-977.

Hicham, B., Ahmed, A. & Mohammed, M. Vehicle type classification using convolutional neural network. 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), 2018. IEEE, 313-316.

Jin, M., Mauch, L. A. & Bienzeisler, B. J. E.-. 2021. A Deep Learning Framework for automated collection and analysis of traffic data based on identifying and classifying Delivery Vehicles in Logistics.

Kausar, A., Jamil, A., Nida, N., Yousaf, M. H. J. a. J. F. S. & Engineering. 2020. Two-wheeled vehicle detection using two-step and single-step deep learning models. 45, 10755-10773.

Li, J., Xu, Z., Fu, L., Zhou, X. & Yu, H. J. T. R. P. C. E. T. 2021. Domain adaptation from daytime to nighttime: A situation-sensitive vehicle detection and traffic flow parameter estimation framework. 124, 102946.

Liang, Q., Zhu, W., Long, J., Wang, Y., Sun, W. & Wu, W. A real-time detection framework for on-tree mango based on SSD network. International Conference on Intelligent Robotics and Applications, 2018. Springer, 423-436.

Ligayo, M. a. D., Costa, M. T., Tejada, R. R., Lacatan, L. L. & Cunanan, C. F. An Augmented Deep Learning Inference Approach of Vehicle Headlight Recognition for On-Road Vehicle Detection and Counting. 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2021. IEEE, 389-393.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B. & Belongie, S. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. 2117-2125.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. & Berg, A. C. Ssd: Single shot multibox detector. European conference on computer vision, 2016. Springer, 21-37.

Liu, Y., Tian, B., Chen, S., Zhu, F. & Wang, K. A survey of vision-based vehicle detection and tracking techniques in ITS. Proceedings of 2013 IEEE international conference on vehicular electronics and safety, 2013. IEEE, 72-77.

Lou, L., Zhang, Q., Liu, C., Sheng, M., Zheng, Y. & Liu, X. Vehicles detection of traffic flow video using deep learning. 2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS), 2019. IEEE, 1012-1017.

Meng, Q., Song, H., Zhang, Y. A., Zhang, X., Li, G. & Yang, Y. J. M. P. I. E. 2020. Video-based vehicle counting for expressway: A novel approach based on vehicle detection and correlation-matched tracking using image data from ptz cameras. 2020.

Rashmi, C. & Shantala, C. Vehicle density analysis and classification using YOLOv3 for smart cities. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020. IEEE, 980-986.

Redmon, J. & Farhadi, A. J. a. P. A. 2018. Yolov3: An incremental improvement.

Song, H., Liang, H., Li, H., Dai, Z. & Yun, X. J. E. T. R. R. 2019. Vision-based vehicle detection and counting system using deep learning in highway scenes. 11, 1-16.

Sudha, D. & Priyadarshini, J. J. S. C. 2020. An intelligent multiple vehicle detection and tracking using modified vibe algorithm and deep learning algorithm. 24, 17417-17429.

Sun, C. & Hu, Y. 2021. Intelligent Highway Traffic Detection Algorithm based on Deep Learning.

Taheri Tajar, A., Ramazani, A. & Mansoorizadeh, M. J. J. O. R.-T. I. P. 2021. A lightweight Tiny-YOLOv3 vehicle detection approach. 18, 2389-2401.

Tsai, C.-C., Tseng, C.-K., Tang, H.-C. & Guo, J.-I. Vehicle detection and classification based on deep neural network for intelligent transportation applications. 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018. IEEE, 1605-1608.

Uijlings, J. R., Van De Sande, K. E., Gevers, T. & Smeulders, A. W. J. I. J. O. C. V. 2013. Selective search for object recognition. 104, 154-171.

Ullah, I. & Lee, H. J. Moving vehicle detection and information extraction based on deep neural network. International Conference on Image Processing, Computer Vision, and Pattern Recognition—IPCV, 2017.

Yang, H. & Qu, S. J. I. I. T. S. 2018. Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition. 12, 75-85.

Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X. & Chen, C. J. I. T. O. I. T. S. 2011. Data-driven intelligent transportation systems: A survey. 12, 1624-1639.

Zuraimi, M. a. B. & Zaman, F. H. K. Vehicle detection and tracking using YOLO and DeepSORT. 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2021. IEEE, 23-29.