An Empirical Research on Vehicle Detection employing YOLOv2 features Transfer Learning and Data Augmentation

V. Sowmya*[1] & R. Radha[2]

[1]Research Scholar, Research Dept. of Computer Science, SDNBV College for Women, University of Madras, Chrompet, Chennai-600044, India.
[2]Associate Professor, Research Dept. of Computer Science, SDNBV College for Women, Chrompet, Chennai-600044, India.
v.sowmy81@yahoo.in[1],radhasundar1993@gmail.com[2]
Corresponding Author: V. Sowmya*

Abstract:

Vehicle Detection and Classification are challenging in the surveillance control and traffic planning realm. In recent years, notably, the performance of Deep Convolutional Neural Networks models for image-based object classification and detection has been significant. From this perspective, in this paper, a study is conducted on Vehicle detection utilizing the YOLOv2 model, enabled with the TensorFlow Python method to enhance image recognition by incorporating the performance features. This model operates on a customized dataset configured with the fine-grained parameters for application in real-time traffic densities. The experiment result manifests the YOLOv2 system features Transfer Learning and Data Augmentation using a custom vehicle dataset achieves significant prediction accuracy, especially with a small object, while meeting with real-time requirements. As a corollary, with the comparison experiments, it demonstrates that the custom YOLOv2 optimized training parameters achieve greater accuracy, mAP of 78.9%, than the traditional and other Convolutional Neural Network models.

Keywords:*Classification, Deep Convolutional Neural Networks, DataAugmentation, TensorFlow, Traffic Surveillance, Transfer Learning, Vehicle detection, YOLOv2.*

1. Introduction

The magnitude of automobiles usage in urban roads has created many challenges to manage and control the traffic. Detection and Tracking of Vehicles applying the Traffic Surveillance system plays a vital role in the management and control of road traffic. The core of Image Processing is Classification, Localization, and Detection of Objects where accuracy, speed, cost, and complexity are key challenges. In the initial phase, the object detection pipeline is classified into three phases: Region Classification, Proposal Generation, and Feature Extraction. Support Vector Machine (SVM) [1] is ordinarily employed for object detection and classification as the model manifests remarkable performance in data-qualification. Additionally, some classification techniques are used in the classification phase such as AdaBoost, bagging, and cascade learning, which leads to further improvements in the accuracy of detection.

In recent years, Deep Convolutional Neural Networks for image-based object classification and detection have achieved remarkable success. Besides, Deep learning methods are commonly implemented in the area of computer vision. Deep neural networks produce hierarchical features in comparison with traditional feature descriptors and various scale information in different layers. Finally, it produces robust and selective features for classification. The object detection frameworks rely on deep learning that is primarily categorized into two families: R-CNN [13] and its variants, YOLO [5] and its variants.

YOLO (You Only Look Once) is an object detection system with real-time effects. Enzeng Dong [6] et al, proposed a system for developing an efficient network structure of YOLOv2 for vehicle data

recognition. Initially, the CNN layers are added then the output size of layers is changed from 13x13 to 26x26 to enhance features from pixel image and the loss function is optimized to adapt the size of an image to improve the detection accuracy. Experimental results show that 79% mAP (mean Average Precision) is achieved for detecting small objects. Aleksa Corovic [7] et al, proposes the newest algorithm YOLOv3 is demonstrated for the detection of congestion population. Five classes are trained and tested in different weather conditions. Experiment results show that 46.60% mAP is achieved. Adami Fatima Zohra [8] et al, focus on the detection and recognition of vehicles in a video stream. Convolutional Neural Network is used in this technique and a dataset that contains images to enable recognition and classification of vehicles. But the improvement in detection speed is lagging. The YOLO algorithm is developed by J. Redmon [17] et al in 2016. This algorithm is used for the detection of objects. The disadvantage of this method is the difficulty of detecting the smallest objects. Experiment results show that 52.7% mAP is achieved. Mengxi.W[18] used the algorithm YOLO to detect vehicles from a video stream dashboard camera. Rachel Huang [19] used the algorithm YOLO-LITE to detect objects in real-time which is configured for non-GPU computers. The model was trained on the PASCAL VOC and COCO dataset. Experiment results show that mAP of 33.81% and 12.26% are achieved.

L.Chen [21] et al proposes a vehicle detection using CNN. Initially, the vehicle scales and aspect ratios in the Jiang Su Highway vehicle Dataset (JHSD) are clustered and trained through CNN. They utilize feature-fusion techniques and also adopt fully convolutional architecture to detect different sizes of vehicles. Experiment results show that 75.7% mAP is achieved. F. Zhang [22] et al proposes a system for vehicle detection with feature concatenation using CNN. The two feature extractors through deconvolution and pooling are used for localization and classification tasks. DP-SSD (Deconvolution Pooling - Single Shot MultiBox Detector) vehicle dataset is constructed and trained through CNN. Experiment results show that better accuracy 75.43% mAP is achieved compared to other datasets. J. Lu [23] et al proposes three public aerial image datasets for vehicle detection and integrated for YOLO training. Experiment results show that 76.7% mAP is achieved. S. Geethapriya [24] et al proposes a vehicle detection using YOLO. The advantages and working of the YOLO algorithm are discussed. The images are predicting by bounding boxes with class probabilities and detect the object faster compared to other algorithms. Jie Liang [28] has trained a YOLO custom model and able to increase the prediction precision accuracy and also detected foreign licence plate with the trained international dataset. Experiment results show that 93% mAP is achieved.

In this paper, a study is carried out using YOLOv2 in the TensorFlow Python method. This research proposes a real-time vehicle detection and classification algorithm. Python [15] has become a common language for many data science and deep learning applications. Besides, OpenCV [16] has become an extensive set of classic and state-of-the-art computer vision and deep learning algorithms [7]. YOLOv2 [4] is an integrated object detection model which is easy to construct and trained directly on full images. Therefore, the entire work in this study is carried out by using YOLOv2 in TensorFlow Python with the OpenCV method especially in the windows platform and implemented in different traffic densities. The system is enabled with core feature of dynamic detection of vehicles. In cities, heavy vehicles are controlled in traffic prone areas by the deployment of computer vision and deep learning techniques in the traffic management systems.

The deep learning algorithms are widely used in the fields of pattern recognition and image processing, precise than traditional algorithms. The main objective of this paper is to enhance accuracy than a CNN-based deep learning method for vehicle detection in real-time settings. In previous papers [21][22], we have found that the accuracy and training of networks are insignificant. To overcome the challenges in the previous method, this paper uses a method called fine-tuning for training a neural network to classify new objects using a subsize dataset. This method freezes a few top layers, unfreeze the base layers, and jointly trains the newly-added classifier layers as well as the final layers of the base model. This mechanism is tuned for improvement of accuracy, reduces time and training speed for custom classes.

The rest of this paper is indexed as follows: Section [2] explains the architecture model, Section [3] explains the proposed methodology. Experimental analysis and the results are given in Section [4] and Section [5] concludes the paper.

2.Architecture Model

The overall architecture model is shown in Figure 1. VGG16 architecture is used as a base network model for vehicle classificationand detectionin the YOLOv2 model. In this work, Transfer learning and Image Augmentation techniques are used to avoid overfitting and to improve training speed. After training the custom vehicle dataset, the classifier, which is built on the VGG16pre-trained model [29], facilitates vehicle classification to identify the vehicle based on the system design. Then the vehicle detection of the model is executed by processing the custom image datasets, and variation of network parameters in the YOLO algorithm. YOLOv2 supports the creation of predicting bounding boxes for each of the detected vehicles. Finally, the computer vision technique is employed to test the images and videos on the trained method. A YOLOv2 model vehicle detection framework is shown in Figure 1.

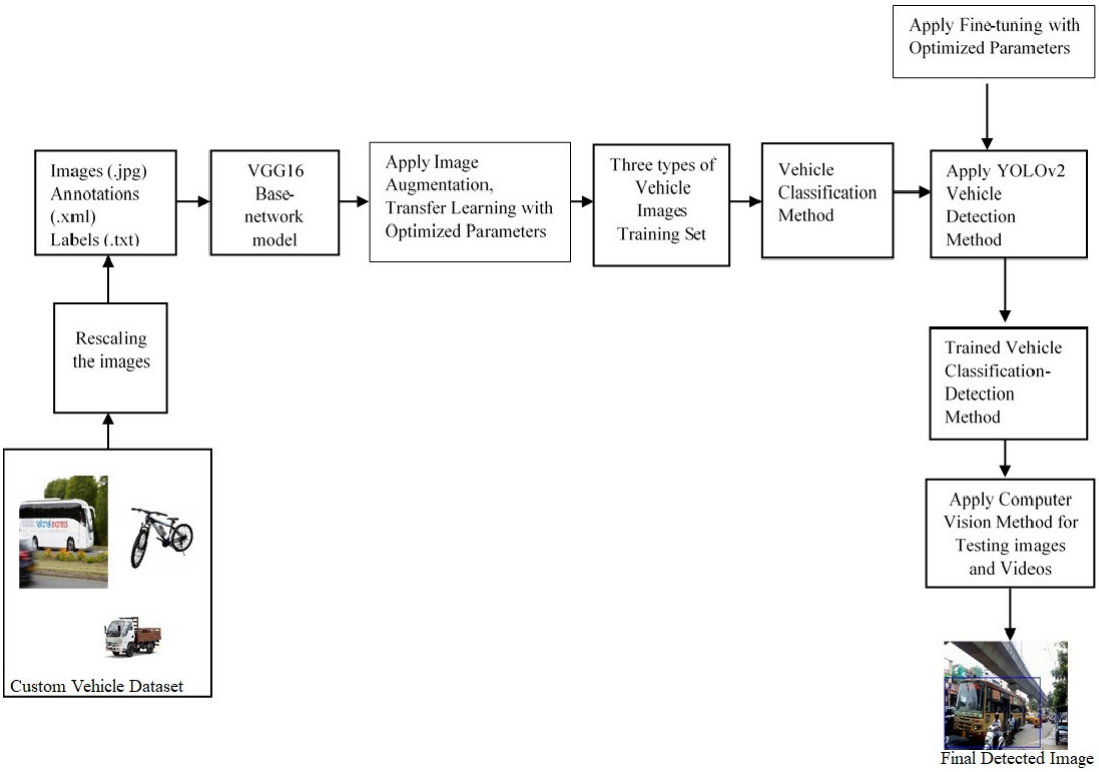

Figure1: A YOLOv2 Vehicle Detection Framework

3. Proposed Methodology

3.1. Base Network

The VGG16 architecture was first introduced by Simonyan, Zisserman in the year 2014. VGG16 model is composed of 13 convolutional layers along with the max pooling, the two fully connected layers, and a SoftMax classifier. In this study, the VGG-16 is used as the base network in YOLOv2 for vehicle detection. The input image size of VGG-16 is 224x224x3 for vehicle classification.The

final layer of the network uses fully connected layers, requires fixed length, which can be done by flattening the output of the last convolutional layer.

3.2. YOLOv2 model

The YOLOv2 model consists of the VGG-16 base network, the fine-tuning convolutional layers, the localizer networks, and the classifier. The model creates a deep, trained end-to-end network on the dataset. Initially, the YOLOv2 model trains the classifier with an image size of 224x224 but retune the classifier with 448x448 for vehicle detection. Further, for a deep network training, batch normalization is critical and should be implemented after the convolutional layer, but before the activation function. The last convolutional layer has 3x3x1024, which concatenates with 13x13x1024 output layers, and finally, the convolutional filter is applied to the new layer 13x13x40 to make predictions. The number of filters in the last layer of the network is calculated as:
Filters= (Class+5) ×5 = (3+5) x5 = 40 filters

3.3. Transfer Learning and Data Augmentation

Transfer Learning and Data Augmentation are used to reduce the over-fitting and to improve the training speed in this proposed model. Transfer Learning customizes the pre-trained model by fine-tuning the optimized parameters while Image data augmentation converts image files into pre-processed tensors to train the model by using various parameters such as rotating, shifting, and flipping images.

3.4. The Loss Function of YOLOv2

YOLOv2 loss function is employed to solve the object detection and classification tasks simultaneously. The stochastic gradient descent optimization method provided by TensorFlow[10] is used with the Adam optimizer[20] to alleviate the cost function. Finally, implement the following loss function, composed of the following terms:

Localization loss - the error between the ground truth value and the predicted boundary box.

$$\lambda_{coord} \sum_{k=0}^{S^2} \sum_{l=0}^{B} 1_{kl}^{obj} (x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2 + \lambda_{coord} \sum_{k=0}^{S^2} \sum_{l=0}^{B} 1_{kl}^{obj} (\sqrt{w_k} - \sqrt{\hat{w}_k}) + (\sqrt{h_k} - \sqrt{\hat{h}_k})^2 \quad (1)$$

From equation (1) 1_{kl}^{obj} equals one if "object appears in cell k." 1_{kl}^{obj} equals one if the "l[th] bounding box predictor in cell k is "responsible" for that detection." λ_{coord} scaling up the weight for the coordinates of the bounding box. Just one bounding box for each object in the ground truth can be liable for that object as shown in Fig 2.

Confidence loss - the objectness of the box.

$$\lambda_{noobj} \sum_{k=0}^{S^2} \sum_{l=0}^{B} 1_{kl}^{obj} (C_k - \hat{C}_k)^2 \quad (2)$$

$\lambda_{noobj}$is the weights of loss when a background is detected.$\hat{C}_k$ is the confidence score of the box l in cell k.

Classification loss - the squared error of the class conditional probabilities for each class.

$$\sum_{k=0}^{S^2} 1_{kl}^{obj} \sum_{c \in classes}^{0} (p_k(c) - \hat{p}_k(c))^2 \quad (3)$$

$p_k(c)$ - conditional class probability for class c.

By using the above loss function, the custom dataset is trained with 75 epochs, and the average loss is 0.82013. The trainingloss and validation loss graph is shown in Figure 2

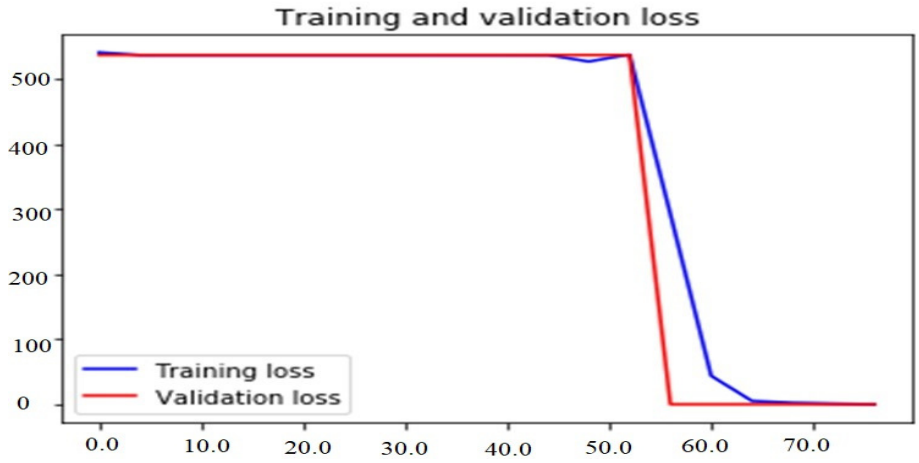

Figure 2: Training and Validation loss of the custom vehicle dataset

4. Experimental Analysis and Results

The deployment and experiment of the YOLOv2 network on the custom dataset are described in this section. An experimental study was conducted in real-time situations of the urban road traffic using video sequences to detect and track the moving vehicles on the road. A total of 50 traffic videos were captured in digital cameras on various traffic densities in Chennai. The videos contain vehicles such as cars, motorcycles, buses, and trucks, etc. A video clip that has a minimum track length of one-minute is selected for vehicle tracking. The detection is possible if there is continuous motion. Then, the experimental results are analyzed from different aspects that demonstrate the performance of YOLOv2.

The proposed system is implemented in Anaconda python 3.6. YOLOv2 enabled with the TensorFlow method is adopted for simulating the Convolutional Network, and OpenCV is used for Image Pre-Processing and also to test the Classifier in real-time. The system environment for simulation is as follows: CPU - Intel Core i5-8265 1.80 GHz, SSD-512GB, RAM - 8GB.

4.1. Dataset

The first step is pre-processing the VEDIC (Vehicle Dataset from Image crawler) custom dataset. The images of all vehicles such as bus, lorry, bicycle are taken from icrawlers. Then rescale the images such that their width and height are 1024x1024. This dataset contains a validation set and a training set. The Vehicle dataset is organized as two separate entities with a ratio of 7:3 where the numbers of images in the training and validation datasets were 1140 and 342, respectively.

4.2. Training a Dataset

This paper uses VGG16 pre-trained model as a base network. To startup with training, the initial weights of pre-trained weights are downloaded. The proposed model can be trained with images and associated annotations. An image file is associated with an annotation file. The custom vehicle dataset for YOLO simulation includes images, annotations, and labels. The input file criteria for images are in .jpeg format, and annotation files are in .xml and labels are .txt formats. The annotation file of each

image contains the class name, GT's (Ground Truth), center point coordinates (x, y, width, height), and coordinates of GT's 4 corners (xmin, ymin, xmax, ymax). For vehicle detection, it generates information for each bounding box. This tool records the appearances of vehicles in each image and the absolute coordinates of the bounding boxes. Training accuracy depends on the drawing in bounding boxes.

The batch size used in the proposed system is 64. The number of epochs is 75. The dataset contains a total of 1140 images. During the initialization of fine-tuning, the YOLOv2 model is loaded with respective weights and .cfg file. The initial learning rate is 0.001, weight decay is 0.0005 and momentum is 0.9. There are some Data augmentation techniques such as shifting and rotating are applied to classify the particular vehicle. To avoid over-fitting in the proposed model each batch normalization layer should be frozen during the training process. Now, train a model with the last four layers that are unfrozen containing convolution layers, max-pooling layers for 30 epochs. Again, unfreeze all the layers for remaining epochs. Therefore, the best weights are saved and used for object detection. An Adam algorithm is used as an optimizer for loss function. A graph of training loss and validation loss is minimized over several epochs as shown in Figure2.

4.3. Performance Metrics

This system defines Mean Average Precision (mAP), Intersection Over Union (IoU), and Confusion Matrix as the performance metrics, which is used to evaluate the performance of the proposed method in the custom vehicle dataset.After training the model, the value of IoU is computed for defining the plot-box to predict the images. It determines the confidence score to evaluate the accuracy of prediction within the trained vehicle data. The higher threshold value (IoU >= 0.7) increases the accuracy or for better object prediction, as shown in Figure3.

The test results of the YOLOv2 model using the custom vehicle dataset are shown in Table I. From Table I(a), the performances of three classes are measured using Precision, Recall, f-score as the evaluation metrics, in addition to the confusion matrix based on True Class and Predicted Class is shown in Figure4. In Table I(b), the custom YOLOv2 model achieves 78.9 mAP with 78.4 recall_score value. Thus, the custom YOLOv2 model is superior based on evaluation metrics.

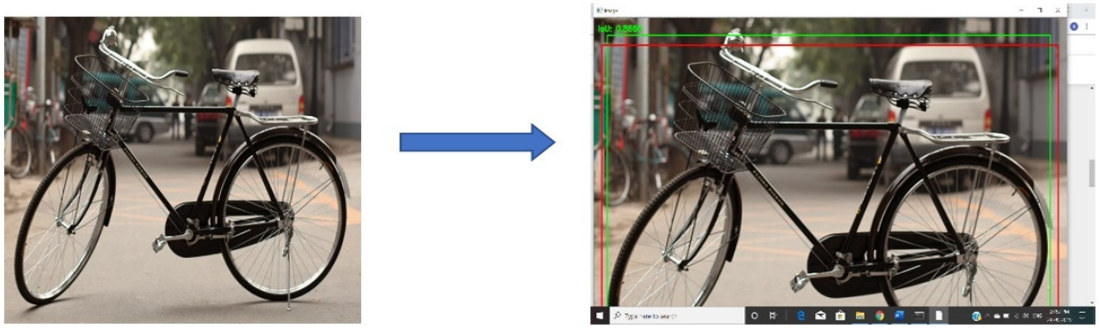

Figure 3: Prediction of IoU value from Original Image

Table I. Test results of the training model using Custom Vehicle Dataset

(a)

Class	Precision	Recall	f-score	Support
Bicycle	0.807	0.800	0.804	110
Bus	0.723	0.892	0.798	111
Truck	0.833	0.661	0.778	121

(b)

Indicator	mAP	Recall_Score	FPS
Value	78.9	78.4	45

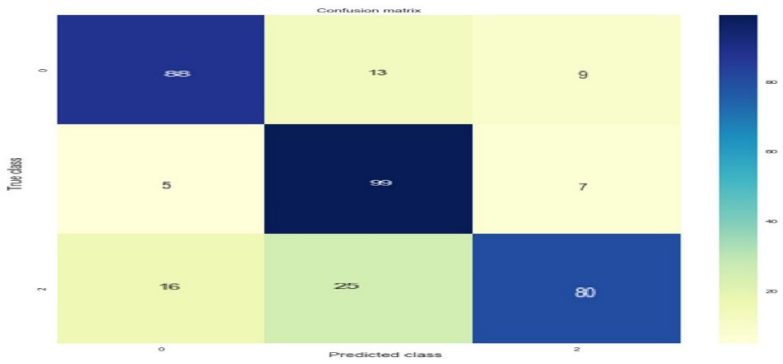

Figure 4: Confusion-Matrix on Custom Vehicle Dataset

4.4. Testing a model

The proposed system is tested to ensure the effectiveness of the module and is compared with the performance of other state-of-the-art approaches is shown in Table II. The classifier is tested by using a computer vision technique. Hence, After the detection and identification of the labeled vehicle in an image, it requires to be tracked to reduce the computation in future frames until the particular vehicle eventually disappears from the video. The tested video is saved as images using computer vision. The classifier was tested on varioustraffic intersections in Chennai, as shown in Figure 5. The video includes all kinds of vehicles such as a car, bus, motorbike, etc. The vehicle detection system is tested in real-time traffic densities in Chennai at various places. A custom dataset is created for the YOLOv2 model and analyzed from the following aspects, Figure 5 depicts the accuracy of the detection of small and large objects. The dynamic-sensing mechanism achieves significant prediction accuracy.

Table II. Comparison Results on Custom Vehicle Dataset with Others

Models	Overall(mAP)	Bus	Truck	Others
JSHD [21]	75.7	91.8	77.3	62.4
DP-SSD [22]	75.43	82.1	70.3	63.8
RON320 [27]	73.6	89.5	75.5	60.2
Our Proposed Model	78.9	72.3	83.3	80.7



Figure 5: Various Vehicle detection using Yolov2 at different junctions (Guindy, Airport, Chrompet) in Chennai

5.Conclusion

A Coherent Vehicle Detection System has been developed, enabled with advanced techniques using Deep Learning and Computer Vision. The proposed system for vehicle detection describes the training procedure of the YOLOv2 custom model. Unlike classifier-based approaches, YOLOv2 - the entire model, is adaptable to training methods comprising full images and a loss function that has effects on detection performance. Finally, the proposed system is tested in real-time applications in Chennai at various places. Further, this study considered Vehicle Detection using YOLOv2 in the TensorFlow python method incorporating Transfer Learning techniques has been successful with significant classification accuracy of 78.9% mAP when compared to the earlier methods achieved through Convolutional Neural Network. Further study, the precision of the algorithm can be explored through the implementation of cross-validation techniques like K-fold cross-validation, and related methods to train our model with large datasets, and work on different weather and lighting conditions.

6.References

[1] Sheeraz Memon, Sania Bhatti, Liaquat Ali and M.B. Talpur. "A Video-based Vehicle Detection, Counting, and Classification System". *International Journal of Image Graphics and Signal Processing*.2018. pp. 34-41.

[2] N. Buch, S.A. Velastin., and A. Orwell. "A review of computer vision techniques for the analysis of urban traffic". *IEEE Trans. on Intelligent Transportation Systems*.2011. Vol. 3 (12), pp. 920–939.

[3] Convolutional Neural Networks. Date accessed: 29/11/2017.https://github.com/pjreddie/darknet.

[4] YOLO- Real-time object detection. Date accessed: 12/03/2018.https://pjreddie.com/darknet/yolo.

[5] Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3. Date accessed: 18/03/2018. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088.

[6] Enzeng Dong, Yanfang Zhu, Yuehui Ji, Shengzhi Du. "An Improved Convolution Neural Network for Object Detection Using YOLOv2". *IEEE International Conference on Mechatronics and Automation*. 2018. pp.1184-1188.

[7] Aleksa Corovic, Velibor Ilic, Sinisa Duric, Malisa Marijan, and Bogdan Pavkovic. "The Real-Time Detection of Traffic Participants Using YOLO Algorithm".2018. Research gate. DOI: 10.1109/TELFOR.2018.8611986

[8] Adami Fatima Zohra, Salmi Kamilia, Abbas Fayçal, Saadi Souad. "Detection and Classification of Vehicles using Deep Learning". *International Journal of Computer Science Trends and Technology (IJCST)*.2018. Vol. 6(3), pp.23-29.

[9] AlexeyAB/darknet.https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects. Date accessed: 29/11/2017.

[10] Trieu. (n.d.), from https://github.com/thtrieu/darkflow. Date accessed: 29/11/2017.

[11] N. Tijtgat. How to train YOLOv2 to detect custom objects. https://timebutt.github.io/static/how-to-train-yolov2. Date accessed: 29/11/2017.

[12] Luc van Gool. Action snippets: "How many frames does human action recognition require?"*IEEE Conference on Computer Vision and Pattern Recognition*.2008. https://doi.org/10.1109/CVPR.2008.4587730.
[13] S. Mukherjee., A. Mallik, D.P. Mukherjee. "Human Action Recognition Using Dominant Motion Pattern". *International Computer Vision Systems. Lecture Notes in Computer Science*. Springer.2015. Vol. 9163 pp. 477-487.
[14] R. Girshick. "Fast R-CNN". *IEEE International Conference*. Computer Vision.2015. pp. 1440–1448.
[15] David Beazley and Brian K. Jones. Python cookbook. 3rd edition,2013.
[16] Michael Beyeler. "Machine Learning for OpenCV". OpenCV library function. Retrieved 12/08/2011.
[17] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. "You only look once: Unified, real-time object detection". *IEEE conference on computer vision and pattern recognition*.2016. pp. 779–788.
[18] W. Mengxi. "Real-time vehicle detection using YOLO". 2017.The Medium website.www.medium.org
[19] Rachel Huang, Jonathan Pedoeem, Cuixian Chen. "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers". 2018. Research Gate, https://arxiv.org/abs/1811.05588.
[20] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". 2014. In: CoRR abs/1412.6980.
[21] Linkai Chen, Feiyue Ye, Yaduan Ruan, Honghui Fan, Qimei Chen. "An algorithm for highway vehicle detection based on convolutional neural network". *in EURASIP Journal on Image and Video Processing*. 2018.
[22] Fukai Zhang, Ce Li, and Feng Yang. "Vehicle Detection in Urban Traffic Surveillance Images Based on Convolutional with Feature Concatenation". 2019. DOI:10.3390/s19030594.
[23] J.Y. Lu, C. Ma., L. Li, X.Y. Xing, Y. Zhang, Z.G. Wang and J.W.Xu. "A Vehicle Detection Method for Aerial Image-Based on YOLO". *Journal of Computer and Communications*. 2018.Vol. 06(11), pp.98-107.DOI: 10.4236/jcc.2018.611009.
[24] S. Geethapriya, N. Duraimurugan, S.P. Chokkalingam. "Real-Time Object Detection with Yolo".*International Journal of Engineering and Advanced Technology (IJEAT)*.2019. pp. 578-581. ISSN: 2249 – 8958
[25] "Object Detection using Yolo" https://www.renom.jp/notebooks/tutorial/image_processing/yolo. Date Accessed: 10/04/2019
[26] Sik-Ho-Tsang.https://towardsdatascience.com/review-yolov2-yolo9000-you-only-look-once-object-detection-7883d2b02a65. Date Accessed: 21/11/2018.
[27] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen. "RON: Reverse Connection with Objectness Prior Networks for Object Detection". CVPR. 2017.pp.5936-5944.
[28] Jie Liang. "Vehicle Detection and License Plate Recognition using Deep Learning". ENSC424 Final Project. http://www.sfu.ca/~jfa49/Files/Vehicle424.pdf
[29] V.Sowmya, R. Radha. "Efficiency - Optimized Approach - Vehicle Classification Features Transfer Learning and Data Augmentation Utilizing Deep Convolutional Neural Networks". International Journal of Applied Engineering Research ISSN 0973-4562 Volume 15, Number 4 (2020) pp. 372-376