ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/350727309

Faster R-CNN and YOLO based Vehicle detection: A Survey

Conference Paper · April 2021

DOI: 10.1109/ICCMC51019.2021.9418274

CITATIONS
88

READS
1,212

3 authors:

Madhusri Maity
Jadavpur University
1 PUBLICATION **87** CITATIONS
SEE PROFILE

Sriparna Banerjee
Jadavpur University
60 PUBLICATIONS **153** CITATIONS
SEE PROFILE

Sheli Sinha Chaudhuri
Jadavpur University
208 PUBLICATIONS **2,435** CITATIONS
SEE PROFILE

All content following this page was uploaded by Sriparna Banerjee on 07 October 2023.

The user has requested enhancement of the downloaded file.

Proceedings of the Fifth International Conference on Computing Methodologies and Communication (ICCMC 2021)
IEEE Xplore Part Number: CFP21K25-ART

Faster R-CNN and YOLO based Vehicle detection: A Survey

Madhusri Maity[1], Sriparna Banerjee[2], Sheli Sinha Chaudhuri[3]
Electronics and Telecommunication Engineering Department
Jadavpur University
Kolkata, India
madhusri.maity@gmail.com[1], sriparnatinni@yahoo.in[2] ,shelism@rediffmail.com[3]

2021 5th International Conference on Computing Methodologies and Communication (ICCMC) | 978-1-6654-0360-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICCMC51019.2021.9418274

***Abstract*— Automatic moving vehicle detection plays a crucial and challenging role in performing intelligent traffic surveillance. Numerous research projects aiming to perform proper detection and tracking of vehicles have been carried out and the methods designed under these projects have found their uses in various important applications for e.g. to minimize the fatal accidents which mainly occur due to negligence of drivers or due to poor visibility during inclement weather condition or due to improper illumination, etc. At present, several deep neural networks have been proposed for performing object detection. This paper presents a comprehensive review of existing Faster Region-based Convolutional Neural Network (Faster R-CNN) and You look only once (YOLO) based vehicle detection and tracking methods. In this survey, we have divided the existing vehicle detection methods into different groups depending upon the architecture (Faster R-CNN/YOLO) which have been used as the backbone of these designed methods. We have organized the entire survey in chronological order so that interrelations between proposed methods can be highlighted. Apart from performing in depth analyses of the existing methods, we have described the respective architectures of Faster R-CNN, YOLO and their proposed variants in details in this survey for better understanding. We have concluded this paper by listing down the limitations of the existing works and unexplored aspects of this research topic. We have also thrown some light on the future scope of this research area.**

Keywords—Vehicle detection; Faster R-CNN; YOLO; Proposed variants; Survey

I. INTRODUCTION

In recent years, vehicle detection has become a popular topic of research among researchers working in related fields due to its' societal importance. According to the survey, every year a large number of people die worldwide because of the fatal accidents which are mainly caused due to the negligence of drivers or poor visibility during inclement weather conditions, etc.The report [1] published by National Crime Record Bureau's Accidental Death and Suicides in India stated that hundreds of people died mainly in two states of India(Andhra Pradesh and Telangana) in the year 2014 due accidents caused by poor visibility during inclement weather conditions. Another report [2] which is published in the website of U.S. Department of Transportation, Federal Highway Administration based on the data collected over a span of 10 years (2007–2016) by NHTSA also stated that approximately 21% of total annual accidental crashes in U.S. occurred due to poor visibility during inclement weather. According to the data published in [3], it is stated that about 90% of the road accidents in India occur due to negligence of drivers. These data and statistics published in several significant sources clearly state the importance of performing accurate vehicle detection in real world.

In the past decade, numerous methods have been designed for accurate tracking and detection of vehicles. Although the traditional vehicle detection algorithms such as Gaussian mixed model (GMM) [4] give promising results but it fails to perform desirably when illumination changes occur or in the presence of background clutter etc. The deep learning methods have inherent feature extraction capability which makes them much more acceptable to researchers compared to the traditional methods as it minimizes the errors occurring in classification tasks which occur due to erroneous handcrafted feature extraction to great extent. As Convolutional neural networks (CNN) are designed to artificially replicate the functional capabilities of human cognitive system, they give better performances in various computer vision tasks compared to the traditional methods. Hence, in this survey we have focused mainly on deep neural networks like Faster R-CNN and YOLO network based vehicle detection methods.

The remaining portions of this survey is organized in the following order: R-CNN and its' proposed variants is described elaborately in Section 2 and the vehicle detection models which are designed based on Faster R-CNN are discussed in Section 3. The architectures of several versions of YOLO detector are studied in details in Section 4 and the methods which are designed based on these architectures are discussed in Section 5. This survey is finally concluded by listing down the unexplored aspects and future work in this research topic.

II. BRIEF INTRODUCTION TO R-CNN AND IT'S PROPOSED VARIANTS

A. *Region-based Convolutional Network (R-CNN) [5]:*

R-CNN is one of the primary deep neural network which is designed to perform object detection.

R-CNN uses object proposals generated by selective search to train CNN for performing object detection and generating

Authorized licensed use limited to: Robert Gordon University. Downloaded on May 27,2021 at 06:43:13 UTC from IEEE Xplore. Restrictions apply.

2000 candidate boxes. Each candidate box is then warped into fixed size and given as input to the CNN which in turn acts as a feature extractor and produces 4096 dimensional feature as a output. This set of features is fed to the SVM classifier to perform classification. In addition to performing classification R-CNN also predicts four offset values to increase the precision of each bounding box. System overview of R-CNN is pictorially represented in Fig. 1.

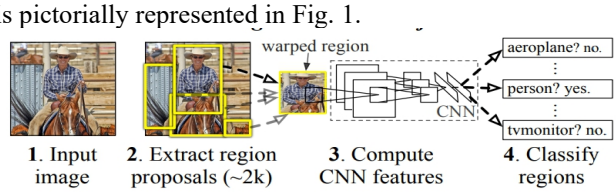

Fig. 1. System overview of R-CNN [5]

Limitations of R-CNN:

i. Huge time complexity, which makes R-CNN not suitable for real life applications.

ii. Inaccurate generation of candidate region proposals due to absence of inherent learning capability in selective search algorithm.

B. Fast R-CNN [6]:

Girshick. et.al. [6] have proposed a modified deep neural network namely, Fast R-CNN to overcome the limitations occurring due to huge time-complexity in R-CNN. Unlike R-CNN, Fast R-CNN does not require to fed 2000 region proposals generated from an image by selective search method to CNN individually to generate corresponding convolutional feature map, instead it feds an entire image as an input to CNN. From the generated feature map of an entire image, region proposals are identified and resized into fixed size by a RoI pooling layer. Then softmax layer is used to identify the objects present within each region proposal and to predict four offset values. The system overview of Fast R-CNN is given in Fig.2.

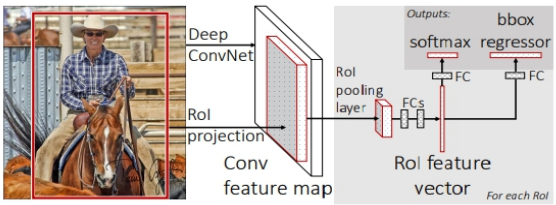

Fig. 2. System overview of Fast R-CNN [6]

Limitations of Fast R-CNN:

i. The introduction of the RoI pooling layer although have reduced the time complexity of Fast R-CNN to some extent compared to R-CNN but the problem of inaccurate region proposals generation occurring due to non-learning capability of selective search algorithm too exists in Fast R-CNN as like R-CNN, in Fast R-CNN too the region proposals are detected using the selective search algorithm.

C. Faster R-CNN [7]:

In order to further reduce the time-complexity and also to generate accurate region proposals, a network namely Faster R-CNN is designed in [7] by merging Fast R-CNN and a novel fully-convolutional neural network namely Region Proposal Network (RPN). RPN not only generates high-quality region proposals but also can simultaneously propose object bounds and objectness scores at each position. The system overview of Faster R-CNN is given in Fig.3.

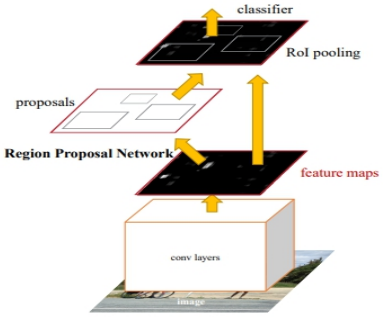

Fig. 3. System overview of Faster-RCNN [7]

Due to the efficiency of Faster R-CNN in performing accurate region proposal generation as well as its capability of reducing the time-complexities of R-CNN and Fast R-CNN to large extent, Faster R-CNN is used by many researchers as the backbone of the deep neural architectures designed by them to perform vehicle detection and tracking in the following section.

III. METHODS DESIGNED BASED ON FASTER R-CNN ARCHITECTURE

1. Fan *et.al.* [8] (2016): In this method, the authors have performed object detection by modifying a few model parameters like training scale, test scale and the number of proposals. The authors in [8] have shown how the performance efficiencies of Faster R-CNN vary in performing vehicle detection using different training scale, test scale and number of proposal values on KITTI dataset [9].

2. Espinosa *et. al.* [10]: In this paper, the authors have performed comparative analyses of performances of AlexNet [11] and Faster R-CNN in performing moving vehicle detection using a video of urban area. They have used Vgg16 [12] as a feature extractor in the Faster R-CNN architecture while performing vehicle detection and finally concluded that Faster R-CNN achieves better F1-score in performing moving vehicle detection.

3. H. Nyugen [13]: In this work, the authors have pointed out that although Faster R-CNN give desirable performances compared to other well-known deep neural networks like R-CNN, Fast R-CNN, AlexNet, etc. but it fails to perform desirably in case of heavy occlusion or large scale vehicle variation or truncation of small vehicles, etc. In order to overcome these shortcomings of traditional model of Faster-RCNN, H. Nyugen has designed an improved architecture of Faster R-CNN where he has adopted MoblieNet

Authorized licensed use limited to: Robert Gordon University. Downloaded on May 27,2021 at 06:43:13 UTC from IEEE Xplore. Restrictions apply.

architecture [14] to build the base convolutional layer of the designed architecture. They have replaced the Non-Maximum-Suppression (NMS) algorithm in Faster R-CNN with a novel algorithm namely, soft NMS. Traditional NMS algorithm checks for all the classes respectively and removes a proposal when the Intersection over Union (IoU) values between the neighbouring boxes for the same class is less than a pre-defined threshold. This property of NMS algorithm often leads to improper vehicle detection when heavy vehicle occlusion occurs. To overcome this drawback, the authors have replaced NMS with soft NMS in their proposed architecture. Soft NMS suppresses proposals based on their objectness scores which are computed according to overlap level of winning proposals and neighbouring proposals and thus reduces the errors occurring in detection tasks. The authors have also substituted RoI pooling layer of Faster R-CNN with context aware pooling layer to fully preserve the contextual information. They have also used the depth-wise convolution structure in MobileNet architecture to perform classification of objects and adjustment of bounding box co-ordinates.

4. Mu *et.al.* [15]: The authors have designed this Faster R-CNN based deep neural network primarily to perform vehicle detection in aerial images. Initially, the authors have performed data augmentation using their proposed oversampling and stitching based data augmentation method in order to solve the discrepancies arising due to small size of vehicles in aerial images as well as to solve the positive and negative samples imbalance issue. The authors here have used ResNet101 [16] as feature extractor in their designed architecture. The traditional model of ResNet101 possess four pooling layers which diminish the feature maps generated for images of size 32×32 into a size of 2×2 , which leads to huge loss of information. So to overcome this information loss, the authors here have performed amplification of feature maps using bilinear interpolation method to preserve the information loss. The authors have also designed a joint loss function by combing the losses of horizontal bounding boxes and oriented bounding boxes so that their designed architecture can detect horizontal and oriented vehicles simultaneously.

IV. YOLO AND ITS' EVOLUTION

A. YOLO version1[17]:

Redmon *et. al.* have designed this object detection network to reduce huge run-time complexities of R-CNN and its' proposed variants. Unlike R-CNN and its' variants, YOLO does not require region proposals to localize and classify objects, instead it divides an entire image into $S \times S$ grid and within each grid it locates $'m'$ number of bounding boxes. Each bounding box predicts a class probability and offset values. The bounding boxes which predict class probabilities below a certain threshold are suppressed. Pictorial representation of steps of object detection using YOLO version 1 is given in Fig.4.

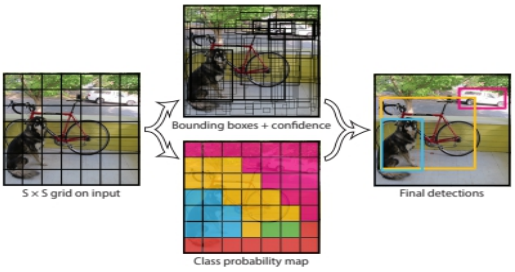

Fig. 4. Steps of object detection using YOLO version 1 [17]

Limitations of YOLO version 1:

i. The maximum number of objects detected by a YOLO detector always depends on the dimension of the grid as YOLO can detect only one object per grid. Like, if the size of the grid is $S \times S$, the maximum number of objects detected is S^2 .

ii. As the maximum number of objects detected by the YOLO detector per grid is 1, so it performs erroneous detection when more than one object exist within a grid.

B. YOLO version 2 [18]:

Redmon et. al. [18] has proposed an improved version of YOLO also known as YOLO9000 which not only excels state-of-art methods like Fast R-CNN, Faster R-CNN in terms of efficiency but also performs detection within a reasonable amount of time. In this version of YOLO detector, the authors have performed various changes in the architecture of YOLO version 1 in order to solve its limitations.

Some notable architectural changes which are done in YOLO version 1:

a. Introduction of Batch Normalization Layer: The introduction of this layer after all convolutional layer improves the performance of the detector and eliminates the chances of overfitting without even adding the dropout layers.

b. Unlike YOLO detector which uses images of dimension 224×224 for training and increases their dimension into 448×448 during test phase. The sudden increase of the image resolution during the test phase decrease the performance efficiency of YOLO detector version 1. Hence, to overcome this drawback YOLO version 2, fine tuning is done and network is trained on images of dimension 448×448 for 10 epochs so that it can gradually adjust with images of high resolution. Hence, the problem arising due to the decrease in mAP (mean Average Precision) which occurs in YOLO due to sudden increase in image dimension is solved.

c. This improved model does not predict the offset values using the fully connected layers which lie on the the top of convolutional layers like YOLO version, instead it removes the fully connected layers from the architecture and predicts objectness scores using the anchor boxes. The use of anchor boxes although reduce the mAP of YOLO version 2 in comparison to YOLO version 1 but it increases its' Recall value.

Authorized licensed use limited to: Robert Gordon University. Downloaded on May 27,2021 at 06:43:13 UTC from IEEE Xplore. Restrictions apply.

d. YOLO version 1 performs training of the network using hand annotated bounding boxes but to make the learning process more easier, the authors in [18] have performed training of their network using bounding boxes generated using k-means algorithm in combination with their proposed distance metric, which is mathematically defined in (1).

$$d(box, centroid) = 1 - IOU(box, centroid) \quad (1)$$

The authors have considered the value of $'k'$ to be 5 in their work as it achieves a good trade-off between the network's performance and complexity.

The other significant characteristics of the model which needs to be mentioned are:

e. Direct location prediction: This characteristic mainly deals with the stability of the method after the introduction of the anchor boxes as the introduction of anchor boxes increases the instability of the model to some extent. To increase the stability of the model, the authors have constrained the co-ordinates of bounding boxes within [0 1] using logistic activation.

f. Fine-grained features: Most of the state-of-the-art methods like Faster R-CNN run on features with different resolutions in order to adapt the network to different resolutions. But in YOLO version 2, instead of running the network on features with different resolutions, the authors have simply added a passthrough layer to the network which concatenates both low resolution as well as high resolution features by adjacently stacking them instead of locating them spatially.

g. Multi-scale Training: Unlike YOLO version 1 which trains network using images of resolution 448×448, YOLO version 2 trains the network using images of different resolutions. This network runs on images of a particular resolution for 10 epochs and then randomly changes the resolution of images. This network has the down-sampling rate of 32, and range of resolutions varies from 302 to 608. This characteristic of the network helps it to adjust to different resolutions and perform efficiently irrespective of image resolutions.

C. YOLO version 3 [19]:

YOLO version 3 is an improved version of YOLO detector which is designed by Redmon et. al. [19]. YOLO version 3 does not use softmax classifier to predict classes of detected objects as it allows the prediction of only one class per object and thus fails to efficiently handle multiclass prediction. To overcome this drawback, YOLO version 3 uses independent logistic classifiers for each class, which allows it to efficiently handle multi-class prediction.

Unlike YOLO version 2 which uses Darknet-19 as feature extractor, YOLO version 3 uses a hybrid feature extraction approach by combining features extracted using Darknet-19 and the residual network. The proposed architecture of YOLO version 3 has several shortcut connections which increases its' performance efficiency while detecting small objects but decreases its' performance efficiency while detecting large and medium objects.

D. YOLO version 4 [20]

YOLO version 4 is designed taking the inspiration from several Bag-of-Freebies and Bag-of-Specials object detection methods. Bag-of-Freebies method increases the inference time and training cost of the detector but increases its' accuracy while Bag-of-Specials methods increases the inference cost of the method to some extent but increases its' accuracy.

Apart from these modifications, other improvements performed in YOLO version 4 model are selection of optimal values of hyper-parameters using genetic algorithms, introduction of data-augmentation methods like Self-Adversarial Training (SAT) and Mosaic, alterations of existing methods like Cross mini-Batch Normalization, Spatial Attention Module, etc.

E. YOLO version 5 [21]

Unlike previous versions of YOLO which have been developed using Darknet research framework, this is the first version of YOLO which is developed in PyTorch framework. This makes YOLO version 5 much more production ready compared to its' previous versions as PyTorch is much more easily configurable compared to Darknet.

Another notable improvement of this version of YOLO is its' run-time. YOLO version 5 is much faster compared to its' previously proposed versions. The inference time of YOLO version 5 is 140 frames per second while inference time of YOLO version 4 is 50 frames per second when it is designed using same PyTorch library as that of YOLO version 5.

V. METHODS DESIGNED BASED ON YOLO ARCHITECTURE

Xu et. al. [22]: In this work, the authors have performed vehicle detection in aerial images using YOLO version 3 network but only after some modifications. The authors have increased the depth of YOLO version 3 network by increasing the number of convolutional layers to 75 as they empirically found that at this depth, the network achieves desirable performance in detecting vehicles in aerial images.

The architecture of YOLO version 3 proposed in [19] cannot detect vehicles in aerial images efficiently due to the small size of vehicles and complex background of images. As the top level features provide more information about small objects, the authors in [22] have mostly modified the connections between up-sampling and down-sampling layers in order to preserve more top level features so that small vehicles can be detected accurately.

2. Ghoreyshi et. al. [23]: The authors have designed two different vehicle detection networks in this work to detect vehicles whose images are taken from Iranian websites. The images of vehicles which are used for training and testing of networks in this work bear a lot of similarities among them. The first network is designed by merging ResNet network [16] and Single Shot Detector (SSD) [24]. ResNet is used for feature extraction and SSD is used for object localization. The second network which the authors have designed for performing vehicle detection in this work inspired by the

Authorized licensed use limited to: Robert Gordon University. Downloaded on May 27,2021 at 06:43:13 UTC from IEEE Xplore. Restrictions apply.

architectures of Vgg network is a modified version of YOLO. Some significant characteristics of the YOLO architecture based network are:

A. Most convolutional layers have filters of size 3x3.

B. The convolutional layers mostly have same number of filters in exception when the size of feature maps be halved .In such cases, the number of filters is doubled to maintain the time complexity of the network.

C. In this network, convolutional layers perform sampling using a stride value of 2.

D. The final layer of the network is a softmax layer where the number of output neurons is equal to the number of output classes.

3. Rahaman et al. proposed a three step real-time wrong way vehicle detection method in [25].

The first step of the method deals with vehicle detection which is done using YOLO version 3 detector [19] which is discussed briefly in Section 4.

The second step of the method deals with tracking of detected vehicles using a centroid tracker. In this step, the bounding boxes of vehicles detected by YOLO version 3 detector in the previous step are fed as inputs and centroid of each bounding box is calculated to detect the current position of the vehicle. This centroid tracker algorithm is designed based on the assumption that the difference in between the position of a vehicle in consecutive frames of a video is very little. Here the tracking method is based on camera view, hence it is done manually. A region of interest is first initialized and a vehicle is only tracked if the computed centroid of its' bounding box lie within the region of interest and then an identification number is assigned to that vehicle and details of the vehicle is entered in the tracking list corresponding to the assigned identification number. Once, the centroid of the bounding box of any vehicle goes out of the region of interest, the details of the vehicle is removed from the tracking list. Also as the vehicle moves, the centroid of its' bounding box changes, then in such cases, the details of the vehicle is updated in the tracking list as long as the centroid of the bounding box of the vehicle lie with the region of interest.

The third step of the algorithm deals with the detection of the direction of vehicles. In [25], the authors have tracked the direction of vehicle using the height of the centroid. Direction of a vehicle is determined in [25] using the following logic:

a. Let when the centroid of the bounding box of any vehicle first comes within the region of interest, then its' centroid height is computed to be H_1 .

b. As the vehicle moves, its' centroid height also gets changed along with its' position. If the updated centroid lies within the region of interest, then its height is computed. Let the computed height is H_2 .

c. If $H_1 < H_2$, then the designed method predicts that the vehicle is coming towards the camera which is considered as the right direction in this work. In other cases, the vehicles are considered to be moving in the wrong direction.

4. Zhou et. al. [26] has primarily designed this method to perform vehicle detection in satellite images. In order to perform vehicle detection using satellite images, in this work the authors have chosen a modified YOLO version 3 network. The modifications are done in YOLO version 3 network considering the fact that the vehicles in satellite images are very small and also the background of satellite images cause interference in performing accurate vehicle detection.

The notable changes done in YOLO version 3 network in order to adapt it to the characteristics of satellite images in this work are listed below:

a. Here the authors have trained the network using an image set comprising of satellite images.

b. In this work, the anchor points are chosen using K-means algorithm. The bounding boxes are generated from those chosen anchor points.

5. Doan et. al. [27] have designed this method to perform vehicle detection and counting using YOLO version 4 network [20] and DeepSORT network [28]. YOLO version 4 network is used in this work to predict co-ordinates of the bounding boxes, class of detected objects and confidence scores of objects. DeepSORT network is used in this work to track detected objects. Kalman filter present in DeeSORT network helps in tracking objects by facilitating the use of previous states to predict the closest frames of objects. It also helps to avoid duplicate tracking of vehicles by setting a threshold in the first frame.

However, as Kalman filter handles each detected object independently, no connection can be established between detected objects and tracked objects. In this work, the authors have solved this problem by using square Mahalanobis distance to combine the uncertainty elements from Kalman filter and Hungarian algorithm to link data.

Counting of detected vehicles is done using YOLO version 4 network in combination with DeepSORT network. In the counting phase, initially the outputs of YOLO version 4 network are fed as inputs to DeepSORT network which in turn assigns an identification number to the vehicle when its' bounding box co-ordinates suggest that it has entered pre-determined region of interest area for the first time, then only the counter corresponding to the object class of that vehicle will be incremented by one.

VI. CONCLUSION, UNEXPOLRED ASPECTS AND FUTURE SCOPE OF WORK

After studying the methodologies proposed in each work we have included in this survey, we can conclude that there is a room for improvement especially from the run-time complexity aspect. In this survey, we have studied several variants of R-CNN and YOLO, but we have found that the existing methods are mostly designed based on the architectures of few of them. So future work in this research area can be focused on designing a vehicle detection method based on YOLO version 5 architecture.

Authorized licensed use limited to: Robert Gordon University. Downloaded on May 27,2021 at 06:43:13 UTC from IEEE Xplore. Restrictions apply.

Apart from performing vehicle detection, another important aspect is to track them properly to prevent collisions. After going through these works, we can conclude that tracking methods should be improved as the tracking methods proposed till date are mostly manual and are solely dependent on camera view.

The similarities between different classes of vehicles as well as small size of vehicles in satellite images also requires fine-tuning of network parameters to achieve desired results.

REFERENCES

[1] M. Ramu (2015) Poor visibility due to bad weather is killing hundreds in accidents. THE HINDU. https ://www.thehi ndu.com/ news/citie s/Hyderabad/poor-visibility -due-to-bad-weather-is killing-hundreds-in-accidents/article743 9794.ece, Accessed 9 Oct 2019

[2] Federal Highway Administration (2018) Road weather Management Program. U.S. Department of Transportation. https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm, Accessed 25 February, 2021.

[3] https://timesofindia.indiatimes.com/india/90-deaths-on-roads-due-to-rash-driving-ncrb/articleshow/61898677.cms, Accessed 25 February, 2021.

[4] C. Stauffer and W. E. L. Grimson, Adaptive background mixture models for real-time tracking, in Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on, Fort Collins, 1999.

[5] R. Girshick, J. Donahue, T. Darrell , J. Malik and UC Berkeley, " Rich feature hierarchies for accurate object detection and semantic segmentation", IEEE Int. Conf. on Computer Vision and Pattern Recognition, USA, June 2014.

[6] R. Girshick, "Fast R-CNN", IEEE Int. Conf. on Computer Vision (ICCV), Chile, December 2015.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", 2015, arXiv:1506.01497v3.

[8] Q.Fan, L.Brown and J.Smith, "A Closer Look at Faster R-CNN for Vehicle Detection", IEEE Intelligent Vehicles Symposium, Sweden, 2016.

[9] A. Geiger,P. Lenz and R.Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite", Conference on Computer Vision and Pattern Recognition (CVPR), USA, 2012.

[10] J.E. Espinosa, S.A.Velastin and J.W.Branch, "Vehicle Detection Using Alex Net and Faster R-CNN Deep Learning Models: A Comparative Study", Int. Visual Informatics Conference, Malaysia, November 2017.

[11] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Communications of the ACM, 60(6), 2017.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", Int.l Conf.on Learning Representations (ICLR), May 2015.

[13] H. Nyugen, "Improving Faster R-CNN Framework for Fast Vehicle Detection", Hindawi Mathematical Problems in Engineering, 2019. Article ID 3808064

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. W., M. Andreetto and H. Adam "MobileNets: efficient convolutional neural networks for mobile vision applications" ,2017, arXiv:1704.04861v1.

[15] N.Mo and L.Yan, "Improved Faster RCNN Based on Feature Amplification and Oversampling Data Augmentation for Oriented Vehicle Detection in Aerial Images", Remote Sensing, 2020.

[16] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", 2015, arXiv:1512.03385v1.

[17] J.Redmon, S.Divvala, R. Girshick and A. Farhadi, "You Only Look Once:Unified, Real-Time Object Detection",2016, arXiv:1506.02640v5.

[18] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger", arXiv:1612.08242v1

[19] J.Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement", 2018, arXiv:1804.02767v1.

[20] A. Bochkovskiy, C-Y.Wang,H-Y.Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", 2020, arXiv:2004.10934v1.

[21] G.Jocher, https://github.com/ultralytics/yolov5, 2020.

[22] B. Xu, B.Wang and Y.Gu, "Vehicle Detection in Aerial Images Using Modified YOLO", IEEE Int. Conf. on Communication Technology, China, 2019.

[23] A.M. Ghoreyshi, A. AkhavanPour and A. Bossaghzadeh, " Simultaneous Vehicle Detection and Classification Model based on Deep YOLO Networks" , Int. Conf. on Machine Vision and Image Processing, Iran, 2020.

[24] W. Liu , D. Anguelov , D. Erhan , C. Szegedy , S. Reed , C-Y. Fu1 and A. C. Berg, "SSD: Single Shot MultiBox Detector", 2016, arXiv:1512.02325v5.

[25] Z. Rahman, A. M. Ami and M. A. Ullah, "A Real-Time Wrong-Way Vehicle Detection Based on YOLO and Centroid Tracking", IEEE Region 10 Symposium (TENSYMP), June 2020.

[26] L. Zhou, J. Liu and L. Chen, "Vehicle detection based on remote sensing image of YOLOv3", IEEE Int. Conf. on Information Technology, Networking, Electronic and Automation Control, June, 2020.

[27] T-N. Doan and M-T. Truong, "Real-time vehicle detection and counting based on YOLO and DeepSORT ",IEEE Int. Conf. on Knowledge and Systems Engineering , Vietnam, 2020.

[28] F. Yu, W. Li, Q. Li, Y.Liu, X. Shi and J. Yan, "POI: Multiple Object Tracking with High Performance Detection and Appearance Feature", 2016, arXiv:1610.06136v1.

View publication stats
Authorized licensed use limited to: Robert Gordon University. Downloaded on May 27,2021 at 06:43:13 UTC from IEEE Xplore. Restrictions apply.