Object Detection and Tracking using YOLO v3 Framework for Increased Resolution Video

Shaikh Shakil Abdul Rajjak, Abdul Kadir Kureshi

Abstract**: *The proposed system is used for vehicle detection and tracking from the high-resolution video. It detects the object (vehicles) and recognizes the object comparing its features with the features of the objects stored in the database. If the features match, then object is tracked. There are two steps of implementation, online and offline process. In offline process the data in the form of images are given to feature extractor and then after to the trained YOLO v3 model and weight files is generated form the pre-trained YOLO v3 model. In online phase, real-time video is applied to feature extractor to extract the features and then applied to the pre-trained YOLO v3 model. The other reference to YOLO v3 model pre-trained is the output of weight file. The YOLO v3 model process on the video frame and weight file extracted features, the model output is classified image. In YOLO v3 Darknet-53 is used along with Keras, some libraries with OpenCV, Tensor Flow, and Numpy. The proposed system is implemented on PC Intel Pentium G500, 8GB and operating system Windows 7 is used for processing our system. The system is tested on PASCAL VOC dataset and the results obtained are accuracy 80%, precision 80%, recall 100%, F1-Score 88%, mAP 76.7%, and 0.018%. The system is implemented using python 3.6.0 software and also tested using real-time video having 1280x720 and 1920x1080 resolutions. The execution time for one frame of video having resolution of 1280x720 (HD) and 1920x1080 (FHD) and 1280x720 (HD) are 1.840 second and 4.414808 seconds respectively with accuracy is 80%.***

Keywords**: *About four key words or phrases in alphabetical order, separated by commas.***

I. INTRODUCTION

The detection of stationary or moving targets and tracking them on real-time video streams is a very important and challenging task to protect fields from enemies. The enemies can be a human, an animal or even an object. is protected by drones or stationary sticks that include device detection and tracking. Video surveillance is a very broad area of research in computer vision applications that continues to identify, recognize and track targets. Object detection compromises and finds the location of objects in the frame of a video sequence.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
Shaikh Shakil A.*, Department of Electronics Engineering, Pravara Rural Engineering Colege, Loni, India, 9689850728, (e-mail: shakils68@rediffmail.com).
Dr. A. K. Kureshi,, Director, Maulana Mukhtar Ahmad Nadvi Technical Campus, Malegaon, India, (e-mail: akkureshi@rediffmail.com).

Every tracking method requires a detection method in every single frame. Object tracking is the process of following one or more objects found on the detection process using a camera. Background subtraction is the most common detection method used by simple object trackers. It is based on comparing two successive frames. The mean shift method and cascade classifier follow this. OpenCV libraries provide us the same approach and have some RGB color detection algorithms. Smart transport has been highly developed to help solve urban traffic problems and fix proven weaknesses, such as lack of sufficient vehicle information and low vehicle information reliability. Scientists worldwide research the identification of vehicles as an important part of this process [8].

Throughout recent years, with the rapid development of information technology, smart transport systems have become an important means of modern traffic management and an unavoidable phenomenon. Vehicle detection as the primary technology for intelligent transport systems is the basis for many important functions, such as traffic flow and density measurement and statistics, vehicle location and tracking, and traffic data mining, etc. [11]. Vehicle detection is one of the primary applications of intelligent transport systems for object detection. This aims at extracting specific vehicle-type data from vehicle-containing images or videos. A new vehicle detection model YOLO V3 with Keras, Numpy, and Tensor flow, OpenCV for image processing and displaying the results is proposed to solve the problems of existing vehicle detection, such as lack of vehicle type identification, poor detection accuracy and slow speed.

II. RELATED WORK

Xun Li et al. [1] introduced the target detection algorithm YOLO v2 and proposed a new network called YOLO-VOC RV. The enhanced YOLO-VOC RV model has an average traffic density accuracy of more than 90 percent Unlike the YOLO9000, YOLO-VOC and YOLO v3 models, it can be seen that the YOLO-VOC RV loses considerably less Recall efficiency when it gets better Accuracy. The final detection results show that an optimized method is more effective for the recognition in multiple targets of different traffic densities. Free flow error rate is 1.4%, false detection rate is only 3.7%, and blocking flow accuracy rate is 96.3%.

Shuoyuan Xu et al. [2] proposed a multi-object detection and tracking algorithm focused on the development and implementation of computer vision. Develop and implement the deep learning object detection YOLOv2 and JPDA, multiple object tracking algorithm.

The quality is quantitatively and qualitatively measured on both public data sets of a traffic surveillance video and UAV flight footage. Compared to a state-of - the-art multiple object tracking algorithm, the performance evaluation of JPDA on public data sets had a performance decline of approximately 20 percent. However the processing speed was much higher, from 6.8 FPS to 37 FPS, respectively. The aerial video test shows that the pedestrian and road vehicles can be identified and tracked, while the tracking algorithm can compensate for a short detection loss time. The limitation of this algorithm when applying UAV visual object detection and tracking is that the unmolded motion of the UAV can lead to false tracking of objects. While implementing UAV visual object detection and tracking, the drawback of this algorithm is that the UAV's unmolded motion will lead to a false tracking of objects.

Rachel Huang et al. [3] emphasizes on YOLO-LITE, a real-time object recognition system designed to run on handheld devices such as a laptop or a mobile phone without the need of a graphics processing unit (GPU). The model was first trained using the PASCAL VOC dataset compared to the COCO dataset, resulting in a mAP of 33.85 percent and 12.36 percent. When implemented with just 7 layers and 482 million FLOPS, YOLO-LITE can operate at approximately 22 FPS on a non-GPU computer and 10 FPS. This speed is 3.78 times higher than SSD Mobilenet, the latest state-of - the-art platform. Based on the original object recognition algorithms YOLO v2, YOLO LITE was intended to create a simpler, faster and more efficient framework to increase usability to a variety of real-time object recognition tools.

Denis Vorobjov et al. [4] proposed a high-resolution video algorithm for more effective detection of small objects. The study of state-of-the-art algorithms for high-resolution video processing algorithms is carried out for this purpose, which can be incorporated in current surveillance systems. For high resolution video processing, the algorithm is based on CNN and consists of the following steps: each video frame is divided into overlapping blocks; object detection is performed in each block with CNN YOLO; post-processing is performed in each block for extracted objects and merging neighboring regions with the same probability class. The suggested algorithm show better performance for detecting small objects on high-resolution images than the common YOLO algorithm in application. The algorithm obtained 35 per cent higher results for 4 K video processing than the YOLO algorithm. NVIDIA GeForce GTX 1070 has been optimized for PCs with key parameters CPU i7 4.3 GHz, RAM 32 Gb, one 4 K resolution frame at 2.64 s.

Franz Franchetti et al. [5] suggested a pipeline method to be incorporated using a two-stage estimation of each rough and refined resolution of each image or video frame to reduce the total number of estimates needed. For both stages the fast object detection model YOLO v2 is used. The model was developed in such a way that the work would be spread through GPUs. The technique proposed maintains high accuracy on 4 K video with an average performance of 3 to 6 fps and 2 fps on 8 K video. Samira Karimi Mansour et al. had suggested a two-step approach to this problem [6]. In the first step, the goal is to provide a mechanism to optimize memory use and in the second step, a multi-threaded approach has been implemented using YOLOv3 to perform real-time object detection on multiple, simultaneous, live streams on a single GPU. In this method, GPU resources are used to full effect. The proposed solution is evaluated on a public dataset and the result shows performance improvements in Central Processing Unit (CPU) use by an average of 13 percent and by 14 percent in frames per second (FPS) compared with YOLOv3. Recently version 3 (YOLOv3) approach has been introduced as a more effective object detection solution, You Look Only Once. Although YOLOv3 can deliver results faster and more accurate than other approaches, it needs to be used in a single powerful GPU system. Occasionally, however, it is necessary to process many real-time object tracking algorithms simultaneously on a single GPU, where each object tracking algorithm receives a live stream from a camera. Detection of the concurrent events on a single GPU from live streams is difficult. In 720p, the FPS value of YOLOv3 and MvcYOLO is 28, while that number drops steeply to 26 on YOLOv3 approach, and continues to decline but slowly to 13 in 4k video quality. In contrast, the FPS value of MvcYOLO decreased more steeply to 26 in the 1080p and 4k image quality. Planning to develop MvcYOLO as potential work in terms of load balancing and network overhead on a large scale camera monitoring environments.

For the YOLO network, Chunsheng Liu et al. [7] suggested an efficient regional proposal extraction method for creating a complete detection structure called ACF-PR-YOLO and using cyclist detection to illustrate the methods. It creates large-scale potential regions that contain objects for the following deep network rather than using the generated area proposals for classification or regression directly as most regional proposal methodology do. ACF-PR-YOLO's conceptual framework consists of three principal parts. First, a regional proposed extraction method based on the aggregated channel function (ACF) is proposed, called the regional proposal method (ACF-PR) based on the ACF. In ACF-PR, ACF is first used to quickly extract candidates and then to consolidate and extend the boundary boxes into applicable regional proposals for the corresponding YOLO network. Additionally, in ACF-PR regional proposals, an appropriate YOLO network was designed for fine detection. Finally, a post-processing step was developed that mapped the YOLO net effects to the original image resulting in detection and localization. In the future, to boost detection efficiency, it is expected to develop an effective detection algorithm that can be adapted to more complex scenarios.

Jun Sang et al. [8] are aiming to improve YOLOv2, a vehicle detection model called YOLOv2 Vehicle. To obtain better anchor boxes, the vehicle bounding boxes on the training data set were clustered with k-means++ clustering, and six anchor boxes of different sizes were chosen. First, the loss feature was improved with standardization to decrease the impact of the different scales of the vehicles. Instead, the YOLOv2 Vehicle Network was planned to increase

extraction efficiency with the multi-layer feature fusion strategy and removal of repeated convolution layers in high layers.

Based on the experimental results, the YOLOv2 Vehicle mAP could hit 94.36 per cent. The model also showed a good ability to generalize using another dataset from the training dataset. Hence, the proposed network is successful in vehicle detection. Through network analysis the ability of VOLOv2 Vehicle to delete features has been demonstrated. While the design suggested in the paper has obtained ideal experimental results, the number of vehicle types and data is relatively small. In future work, more actual vehicle data is collected to further analyze how to improve the accuracy and speed of vehicle detection.

Wu Zhihuan et al. [10] proposed work addresses the issue of rapid object recognition for high-resolution remote sensing images using CNNs. The authors discussed variant of YOLO is used for object detection within high-resolution remote sensing images. Experimentation on the NWPU VHR-10 dataset, the airport data set and the GoogleEarth aircraft data set shows that the YOLO model is widely applicable to remote sensing images, specifically at predictive speed. The key disadvantages of YOLO are its low alignment accuracy, poor training approximation and generalization of images and objects similar to each other in the unusual aspect ratio. It requires a large number of high-quality Ground Truth labels for model training, based on technical research findings and a lot of manual labor. Future research therefore focus is to solve these problems.

Junyan Lu et al. proposed the vehicle detection system based on the YOLO deep-learning algorithm for an aerial image [11]. This method integrates an aerial image data collection ideal for studying YOLO by manipulating three public datasets. The training model has good test results, and meets the needs in real time, particularly for small objects, rotating objects, as well as compact and dense objects. Next, more public aerial image datasets will be introduced to increase the number and variety of training samples while at the same time improving the YOLO algorithm to further improve the accuracy of the detection.

Joseph Redmon et al. [12] made those modifications to YOLO. He's made a bunch of small changes to the design to make it better. It is a bit bigger but more accurate than the last time. It's still fast though, don't worry. At 28.23 mAP YOLOv3 runs at 320x 320 in 23 ms, as accurate as SSD but three times faster. Looking at the old detection metric YOLOv3.5 IOU mAP is pretty good. It achieves 57:87 AP@50 on a Titan X in 51.21 ms, compared with RetinaNet's 57:5 AP@50 in 197.59 ms, similar performance but 3.7 times faster.

Aleksa legenorović et al. [13] developed the YOLOv3 algorithm to identify participants in real-time traffic. Weights of the neural network were initialized using a pre-trained prototype trained on the COCO dataset. The neural network was further educated on the Berkley Deep Drive dataset, specializing in detecting five classes of traffic participants. On the custom dataset, fake detections from photos representing different traffic situations in Novi Sad have been investigated. The key benefit of the solution proposed that YOLO neural network's flexibility in defining and controlling multiple classes of traffic participants in real-time. The solution also provides a real-time response required for the creation of the ADAS components. Implementing the YOLO algorithm set out in this paper provides a solid foundation for object detection method as part of the ADAS. The accuracy of the algorithm could be improved in future work by focusing on the larger and more diverse data sets that cover different weather and lighting conditions. This algorithm could also be used for reducing the number of false detections in fusion with other sensor readings.

Joseph Redmon et al. [14] developed the modern approach of identification of objects, using YOLO. Earlier work on object detection suggests classifier detection. Instead, framed object recognition was the spatially separated bounding boxes and related class probabilities as a regression problem In one study, a single neural network decides bounding boxes and category probabilities directly from full images. Since the whole detection system is a single network, the performance of end-to-end detection can be directly optimized. This streamlined architecture is incredibly fast. The base code YOLO handles pictures in real time at 45 frames per second. Quick YOLO, a smaller network variant, processes a remarkable 155 frames per second while doubling other real-time detectors mAP. Compared to state-of-the-art detection systems, YOLO makes more localization errors, but is less likely to predict false positives on history. YOLO is therefore studying very general representations of objects. While it generalizes from natural images to other fields such as artwork, it outperforms other recognition methods, including DPM and R-CNN.

Shubham Shinde et al. [15] suggested a paradigm for the real-time understanding of human behavior in video based on YOLO. The key finding of this analysis is that even a few frames in a video can be used to identify behavior. Even in some cases a single frame has been considered adequate to discern action. The accuracy obtained by the approach suggested was 88.462%. The future development of the proposed method is centered on the hunt for a metric that can be used to complete the action recognition within a few frames, thereby eventually stopping the classification of the actual action. Continue to focus on improving behavior identification by using frame object detection to identify more complex human actions. Moving objects or Euclidean distance between moving object centers and people can also provide more knowledge about video-related activities.

Jing Tao et al.[16] suggested a YOLO method with the latest YOLO-based CNN which is very efficient while improving other detectors of objects. The two excellent YOLO and R-FCN object detection algorithms are integrated to obtain higher precision efficiency for traffic scene images. This object detection system gets more reliable after being applied to a night image preprocessing procedure. Therefore, the traffic scene provided a fast, accurate, and robust object detection system based on YOLO.

Open Statistics K G Shreyas Dixit et al. [17] discusses about YOLO (You look only once), a prominent representative of CNN who comes up with a totally different strategy of understanding the role of object recognition. YOLO has achieved fast speeds with fps of 155 and mAP of around 78.6, thus greatly exceeding the performance of other CNN models.

Furthermore, YOLOv2 performs an outstanding trade-off between precision and speed related to the recent advances and also as a detector with effective generalization capability to reflect a complete image.

Jiwoong Choi et al. [18] introduced an image processing algorithm that could achieve better accuracy and speed of recognition in real time and control autonomous driving miss localization The suggested method enhances accuracy, increases the TP, and substantially reduces the FP while maintaining real-time capability by Gaussian modeling, reconstruction of loss function, and the use of uncertainty on position. The suggested Gaussian YOLOv3 method improves the mAP datasets of KITTI and BDD by 3.09 and 3.5 compared to the basic line. The suggested algorithm also lowers the FP by 41.30 percent and 40.53 percent respectively, and increases the TP at 7.376 percent and 4.29 percent respectively on the KITTI and BDD datasets. As a result, the suggested algorithm would greatly enhance the camera-based object recognition system for autonomous driving and is intended to contribute significantly to the large use of autonomous driving applications.

Juan Du [19] briefly addresses the latest algorithms for object detection inluding the CNN and the YOLO team. As with CNNs, YOLO has more sophisticated implementations in practice. YOLO is an Integrated Object Detection System. It's easy to construct and can be trained directly on complete images. Unlike classifier-based methods, YOLO is trained on a loss function that directly correlates to detection efficiency and training the whole model together Fast YOLO is the fastest object detector for general purposes. So YOLOv2 offers the best trade-off between real-time acceleration and the excellent precision of object detection over a range of data sets than other detection systems.

III. IMPLEMENTATION OF PROPOSED SYSTEM

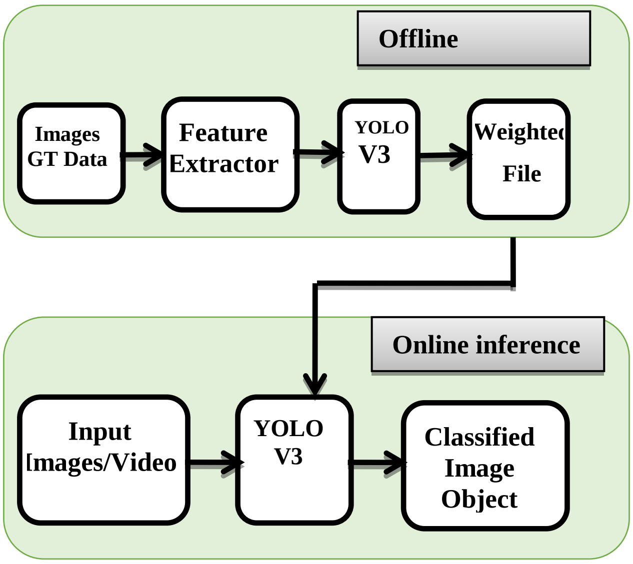

Fig.1. Object Detection and Tracking Proposed

Figure 1 shows implementation of proposed technique for object recognition and tracking using pre-trained YOLO v3 model. There are two processes taking place, online and offline process. The data in the form of images is given to the feature extractor in offline processing and then these extracted features are applied for training to the YOLO v3 system. It creates the YOLO v3 model weight files. In the online process, the output does not know what video or image is on the input side and what the image or video is exactly. The video input is applied to the pre-trained model YOLO v3. The weight file output and the video input are the two data for the pre-trained model YOLO v3. These two pieces of information are processed by the YOLO v3 system to get the classified image as output. Instead of using Keras, we used some more libraries that are OpenCV, Tensor flow and Numpy in YOLO V3 Darknet-53.

IV. ALGORITHM AND FLOWCHART

Figure 2 shows Flow diagram for object detection and tracking using pre-trained YOLO v3 model.

Algorithm step-by-step

Step I: The input video file is given to the module.

Step II: Frames are extracted from the videos.

Step III: Extracted frames are given to the Deep Neural Network.

Step IV: Yolo V3 weight file is also given to the Deep Neural Network (YOLO v3).

Step V: Object (car) is detected here with the help of the YOLO v3 module.

Step VI: Cars are counted hereafter the object is detected.

Step VII: It measures the execution time for the frame per second.

Step VIII: Write to the CSV file.

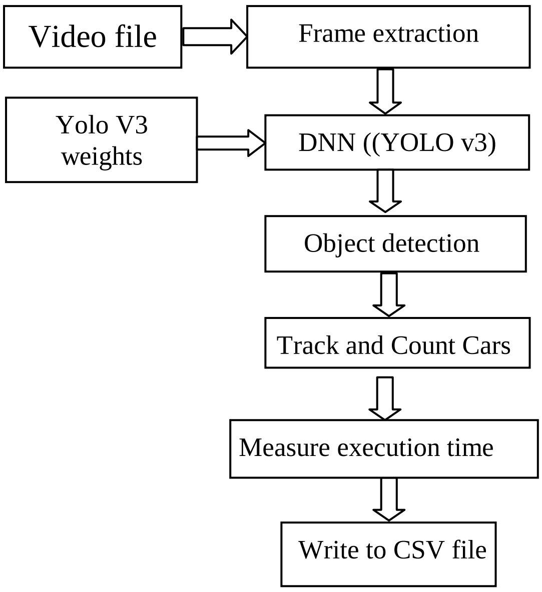

Fig. 2: Flow diagram for object detection and tracking using pre-trained YOLO v3 model.

V. EXPERIMENTATION

We use OpenCV with pre-trained YOLO 3,

there are a few reasons we may want to use OpenCV for YOLO:

1. Easy integration with an OpenCV program: If your software already uses OpenCV and you just want to use YOLOv3,

Retrieval Number: E3038039520/2020©BEIESP
DOI: 10.35940/ijitee.E3038.049620
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

you don't have to worry about compiling and creating the additional Darknet code.
2. OpenCV CPU version is 9x faster: OpenCV's DNN module CPU implementation is incredibly fast. For example, Darknet takes about 2 seconds on a CPU to infer on a single image when used with OpenMP. The implementation of OpenCV, on the other hand, runs in just 0.22 seconds.
3. Support for Python: Darknet is written in C and does not support Python officially. OpenCV, on the other hand, does. There are, however, python ports for Darknet..

A. Datasets used for Proposed Method

The publicly available PASCAL VOC database will be used for the planned implementation. It consists of 10k annotated images with 20 classes of objects with annotations of 25k objects (XML). These images are downloaded from the flicker. This dataset is used in the since 2006 year-round PASCAL VOC Challenge.

B. Implementation Details

This work was carried out in python3. Keras is used to train the image processing network and OpenCV. The system specifications for which the model is trained and evaluated are as follows: Intel Pentium G500 CPU PC, 8 GB RAM.

C. Framework Used

The architecture of YOLO v3 is much more complex than that of YOLO v2, and its detection on small objects as well as compact dense objects or strongly overlapping objects is solid. The principal features of YOLOv3 are as follows:
1. YOLO v3 replaces Softmax Loss YOLO v2 with Logistic Loss. When the predicted object classes are complex, particularly when there are several overlapping labels in the dataset, the use of Logistic Regression is more efficient.
2. YOLO v3 uses nine anchors instead of the five YOLO v2 anchors, thus raising the IoU.
3. YOLO v2 uses only one detector while YOLO v3 uses three which increases the detection effect on small objects substantially.
4. YOLO v3 replaces the darknet-19 network of YOLO v2 with the darknet-53 network, which increases the accuracy of object detection by expanding networks.

D. Video details used for testing the proposed method as shown in Table I

Table I: High Definition (HD) video details

Size	24.7MB
Size on disk	2,59,95.419
Length/duration of video	00:00:27
Frame/image width	1280
Frame/image height	720
Data rate	7700 kilobits per second
Total bitrate	7700 kilobits per second
Frame rate	30.00 frames per second

Table II: Full High Definition (FHD) video details

Size	4.01 MB
Size On Disk	4.01 MB
Length/duration of video	00:00:10
Frame/image Width	1920
Frame/image Height	1080
Data Rate	3232 Kilobits Per Second
Total Bitrate	3360 Kilobits Per Second
Frame Rate	29.00 fPS (frames per second)

VI. RESULTS AND DISCUSSION

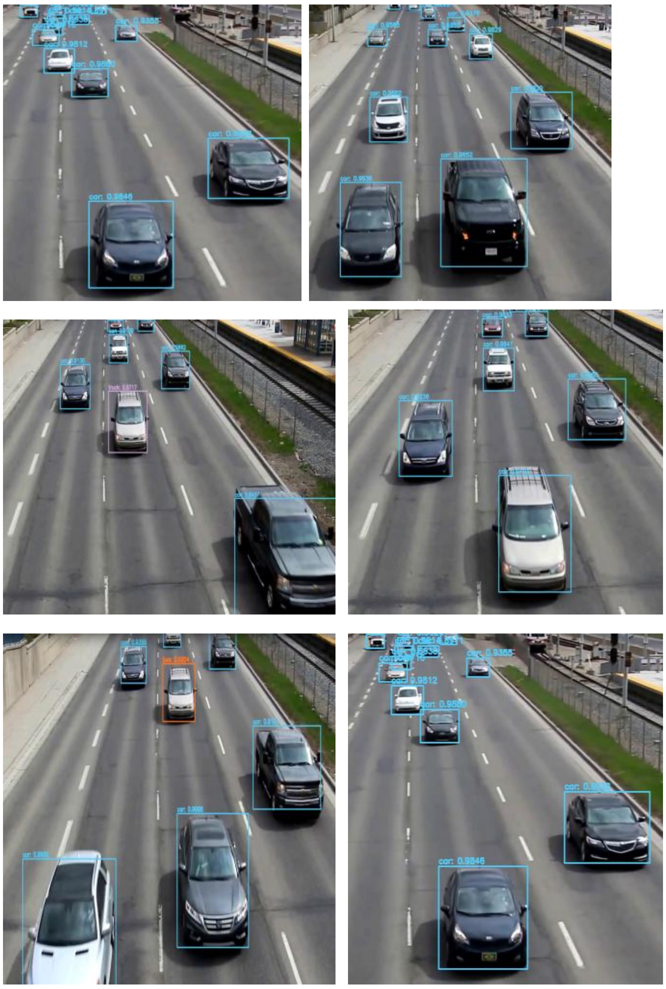

Fig. 3: Showing the object detected is a car and tracked continuously with its probability

Detection of vehicles is the process of identifying a vehicle from a video image / frame with bounding boxes showing its location. A redundant collection of overlapping bounding boxes within the image / frame is suggested as potentially useful areas for the region's proposal network. A trained model file would then attempt to identify the object type in each bounding box. This approach means the classifiers used in traditional object detectors will look multiple times at the same part of an image. YOLO is special, as it looks at a certain part of an object only once, like other classifiers. As an object detector, this technique is much quicker with comparable precision. OpenCV is used to display the Results. We could detect a car in the frame using Yolo v3 and measure the accuracy using the confusion matrix. The above figure shows the detected object is a car with different probabilities and continuously monitored with a bounding box of green color. The various probabilities indicate that the detected object is a car with a probability of 0.82, 0-1 indicates the likelihood of being a car. Figure 3 shows the object detected is car and tracked continuously with its probability. Figure 4 and 6 shows execution for frames (in sec) for HD and Full HD, figure 5 and 7 shows number of cars detected per frame.

Table III shows result for proposed method tested on PASCAL VOC dataset and table IV and figure 8,9, 10 shows comparison of suggested technique with recent techniques for mAP, test time (Sec/Frames) and rate (Fps) (Tested on PASCAL VOC2012 Dataset) (Reference paper 20). Table V shows comparison of suggested method with recent techniques concerning accuracy.

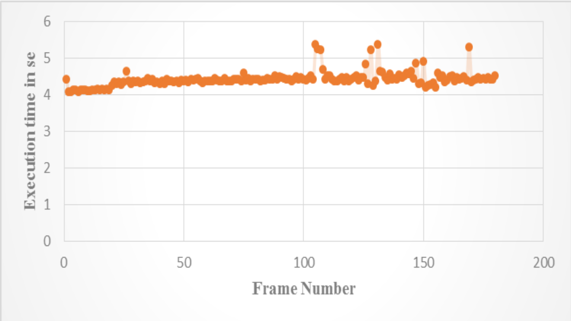

Fig. 4: Execution time for HD video

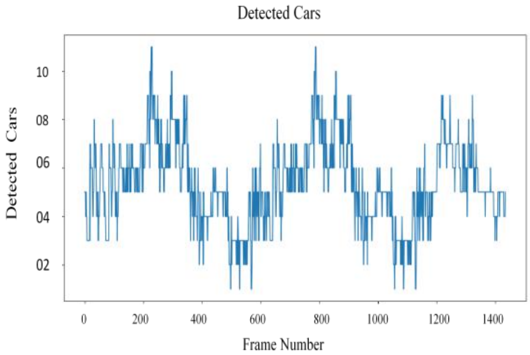

Fig 5: Number of Cars Detected from HD video

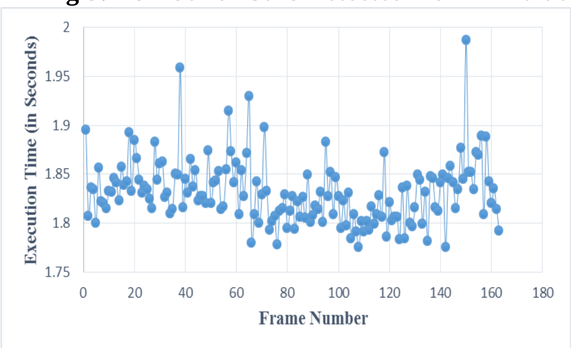

Fig 6: Execution time for FHD video

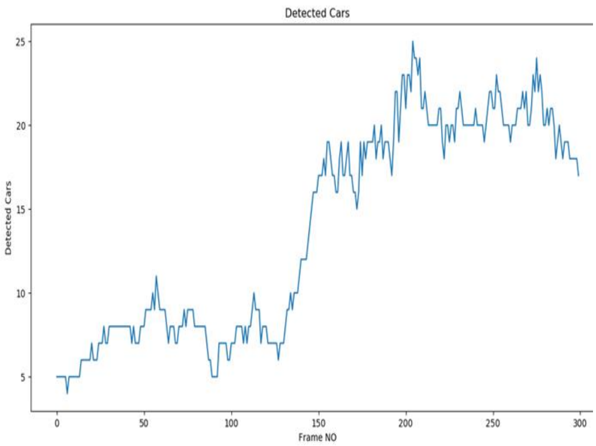

Fig 7: Number of Cars Detected from FHD video

Table III: Result for Proposed Method Tested on Pascal VOC Dataset:

Sr. No.	Parameters	Parameter
1	Accuracy	80%
2	Precision	0.8
3	Recall	1
4	F1-Score	0.88
5	IOU	100%

Table IV: Assessment of Suggested Method with Recent Technique for mAP, Test Time (Sec/Frames) and Rate (Fps) (Tested on Pascal Voc2012 Dataset) (Reference Paper 20)

Methods	mAP (%)	Execution Time Sec./Frame	Rate (fps)
SS+FRCN	66.9	1.72	0.6
SS+HYPERNET*	76.3	0.20	5
Faster R-cnn (vgg 16)	73.2	0.11	9.1
Faster R-cnn (Resnet 101)	83.8	2.24	0.4
YOLO	63.4	0.02	45
SSD 300	74.3	0.02	46
YOLO v2 (544x544)	78.6	0.03	40
YOLO v3(416x416) our work	76.7	0.018	55

Table V: Assessment of Suggested Method With Recent Techniques Concerning Accuracy

Methods	Faster R-CNN	R-FCN	SSD @512	YOLO @554	R-FCN **	Our method YOLO v3 @416
Accuracy	70.4	77.6	74.9	73.4	80.5	80

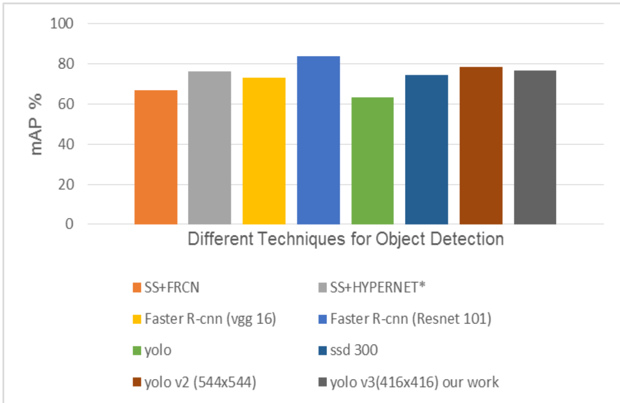

Fig. 8: Comparison of Proposed Method with Recent Method Concerning MAP (%)

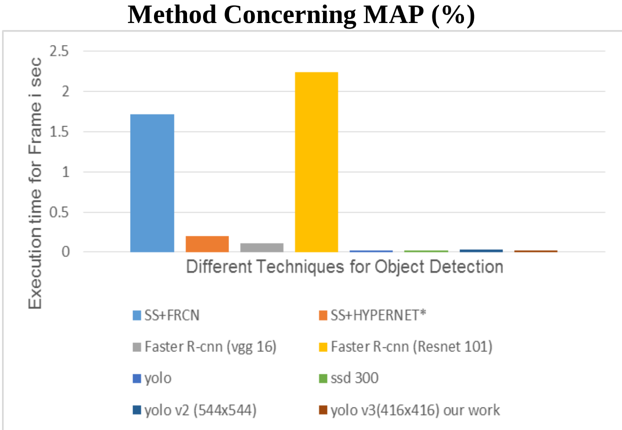

Fig. 9: Assessmnet of Suggested Method with Recent Method Concerning Execution Time Sec./ Frame

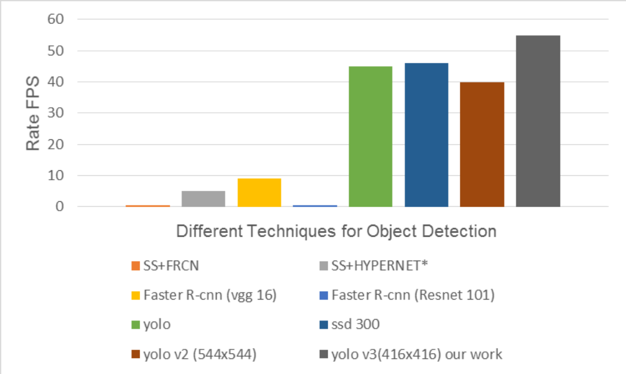

Fig. 10: Assessment of Suggested Method with Recent Method Concerning Frame Rate (FPS)

VII. CONCLUSION AND FUTURE WORK

The proposed system is used for vehicle detection and tracking from the high-resolution video. The system uses the pre-trained YOLO v3 framework to detect and track the object with Keras, Numpy, Tensor flow, and OpenCV. For object detection and tracking, there are two phases offline and online processing The pre-trained template YOLO v3 is trained with some vehicle images and is tested using the dataset PASCAL VOC. To detect and monitor the object in video, real-time video is applied to the pre-trained YOLO v3 model in the offline step. It identifies the object and identifies the object by comparing its features to the features of the objects stored in the database. If the features match, the object will be tracked. The proposed system is implemented on PC Intel Pentium G500, 8GB RAM and Windows 7 operating system. When the system is tested on the PASCAL VOC dataset and the results obtained are accuracy 80%, precision 80%, recall 100%, F1-score 88%, mAP 76.7% and 0.018 seconds execution time per frame. The system is also tested with a real-time video of 1280x720 (HD) and 1920x1080 (FHD). The average execution time for 1280x720 (HD) and 1920x1080 (FHD) video is 1.840 seconds and 4.414 seconds respectively with an accuracy of 80%.

The limitation of the proposed detection of moving objects is that it cannot be used in real-time applications. Further development of a proposed system to detect with improved speed and accuracy will be the future scope.

REFERENCES

1. Xun. Li, Yao. Liu, Zhengfan. Zhao, Y. Zhang, and L. He, “A Deep Learning Approach of Vehicle Multitarget Detection from Traffic Video,” Journal of Advanced Transportation, vol. 2018, pp. 1–11, Nov. 2018.
2. Shuoyuan. Xu, A. Savvaris, S. He, H. Shin, and Antonios Tsourdos, “Real-time Implementation of YOLO+JPDA for Small Scale UAV Multiple Object Tracking,” 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Jun. 2018.
3. Rachel Huang, J. Pedoeem, and Cuixian. Chen, “YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-Graphical Processing Unit Computers,” 2018 IEEE International Conference on Big Data (Big Data), Dec. 2018.
4. Dnise Vorobjov, Iryna Zakharava, R. Bohush, and S. Ablameyko, "An Effective Object Detection Algorithm for High-Resolution Video by Using CNN," Lecture Notes in Computer Science, pp. 503–510, 2018.
5. Vit Ruzicka and Fanz. Franchetti, "Fast and accurate object detection in high-resolution 4K and 8K video using Graphical Processing Unit s," 2018 IEEE High-Performance Extreme Computing Conference (HPEC), Sep. 2018.
6. S. K. Mansoub, R. Abri and Anıl Hakan Y.ı, "Concurrent Real-Time Object Detection on Multiple Live Streams Using Optimization CPU and Graphical Processing Unit Resources in YOLOv3" SIGNAL 2019: The Fourth International Conference on Advances in Signal, Image, and Video Processing, pp. 23-28
7. Chunsheng Liu, Yu Guo, S. Li, and Faliang Chang, "ACF Based Region Proposal Extraction for YOLOv3 Network Towards High-Performance Cyclist Detection in High-Resolution Images," Sensors, vol. 19, no. 12, p. 2671, Jun. 2019.
8. Jun Sang, Zhongyuan Wu, P. Guo, Haibo Hu, H. Xiang, Q. Zhang, and B. Cai, “An Improved YOLOv2 for Vehicle Detection,” Sensors, vol. 18, no. 12, p. 4272, Dec. 2018.
9. Bilel. Benjdira, Taha. Khursheed, Anis. K., Adel A, and K. Ouni, “Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3,” 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Feb. 2019.
10. W. Zhihuan, Xiangning C, Y. Gao, and Yuntao. Li, "Rapid Target Detection in High-Resolution Remote Sensing Images using Yolo Model," ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII–3, pp. 1915–1920, Apr. 2018.
11. Junyan. Lu, Chi. M., Li Li, X. Xing, Yong Z., Zhigang W and J. Xu, “A Vehicle Detection Method for Aerial Image Based on YOLO,” Journal of Computer and Communications, vol. 06, no. 11, pp. 98–107, 2018.
12. Joseph R. and Ali F., “YOLOv3: An Incremental Improvement”, University of Washington, 2018
13. Alexa Corovic, Velibor. I, Sinisa D, M. Marijan, and B. Pavkovic, “The Real-Time Detection of Traffic Participants Using YOLO Algorithm,” 2018 26th Telecommunications Forum (TELFOR), Nov. 2018.
14. Joseph R, S. Divvala, Ross G, and Ali. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2016.
15. Shubham. Shinde, Ashwin K and Vikram G, “YOLO based Human Action Recognition and Localization,” Procedia Computer Science, vol. 133, pp. 831–838, 2018.
16. Jing. T., Hongbo Wang, X. Z, Xinyu. Li, and H. Yang, “An object detection system based on YOLO in traffic scene,” 2017 6th International Conference on Computer Science and Network Technology , Oct. 2017.
17. K G Shreyas Dixit, Mahima Girish Chadaga, Sinchana S Savalgimath, G Ragavendra Rakshith, Naveen Kumar M R "Evaluation and Evolution of Object Detection Techniques YOLO and R-CNN," Special Issue, International Journal of Recent Technology and Engineering vol. 8, no. 2S3, pp. 824–829, Aug. 2019.
18. J. Choi, D. Chun, Hyun K, and Hyuk-Jae Lee, “Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving” arXiv:1904.04620v1 [cs.CV] 9 Apr 2019, pp 1-8
19. Juan. Du, “Understanding of Object Detection Based on CNN Family and YOLO,” Journal of Physics: Conference Series, vol. 1004, p. 012029, Apr. 2018.
20. Zong.-Q. Zhao, Peng. Z, Shau.-T. Xu, and X. Wu, “Object Detection With Deep Learning: A Review,” IEEE Transactions on Neural Networks and Learning Systems, pp. 1–21, 2019.
21. https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359

AUTHORS PROFILE

Mr. Shakil A. Shaikh is pursuing his Ph. D and working as Associate Professor in Electronics Engineering at Pravara Rural Engineering College Loni, Ahmednagar, Maharashtra, India. He has more than 29 Years of teaching experience. His area of interest is Signal and Image processing He is a Life Member of ISTE. He has also applied for Fellow member of IETE. Currently working with Colab for the research work.

Dr. A K Kureshi had completed his Ph. D in 2010 in the area of Low Power Techniques and Architecture of FPGAs. He is working as a Director at Maulana Mukhtar Ahmad Nadvi Technical Campus, Malegaon, Maharashtra, India. He has more than 22 years of teaching experience. His area of interest is in the field of Carbon Nano Tubes as Interconnects and logics and in Multi-layer Graphene domain. He is a Member of IEEE and Life Member of ISTE, IEI and IETE