JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Ensemble Based Deep Network for Vehicle Detection, Classification and Speed Estimation

[1]Madhuri Mane, [2]Rashmi Borase, [3]Vaishnavi Kulkarni, [4]Omkar Nikhal, [5]Sumeet Singh

[1]Assistant Professor, [2,3,4,5]Student
[1,2,3,4,5]Department of Computer Engineering
[1,2,3,4,5]SCTR's Pune Institute of Computer Technology, Pune, Maharashtra, India

Abstract: Speed detection and analysis of traffic data is a challenging task that plays a vital role in the safety of civilians considering the dangers imposed with everyday transportation. The presence of an automated and intelligent traffic surveillance system in today's civilized society is of critical importance for monitoring traffic activity and detecting overspeeding vehicles. We present an accurate and effective computer vision-based system for monitoring vehicular traffic by detecting and classifying vehicles and extracting crucial statistics such as average road speed, overspeeding detections, vehicle types, frequency of vehicles, etc. Our proposed system implements a machine learning-based approach that makes use of computer vision techniques and classifiers on real-world traffic video surveillance for effective vehicular detection and classification as well as for obtaining traffic statistics.

***Index Terms* - Vehicle Speed Estimation, Computer Vision, Traffic Video Analysis, Vehicle Detection, Machine Learning.**

I. INTRODUCTION

The concept of Machine Learning is to create a system that automatically learns and evolves without being explicitly programmed. Humans have applied this method to various fields of technology, research, and engineering to achieve systems that can determine future weather, convert a person's voice to text, predict a quantity's value from other values that it seems to depend upon, recognize objects, texts, and faces in an image, etc. These tasks are easy for humans to perform, and now we can program machines to do the same. One system mentioned above is "Image Recognition": The task of recognizing and identifying entities that are present in an image. These entities may include physical objects, text, numbers, faces, patterns, etc. This model has various applications from facial recognition for security to text recognition for simple convenience.

One such scenario where Image Recognition is applicable is traffic and vehicles. Given a big set of images of vehicles, a system can be trained to detect vehicles in more images. Similarly, the system can learn to detect objects that are typically seen with vehicles like roads and traffic lights. Such a system would be useful because it will enable a vast array of analyses and computations to be done on the images. This can be further expanded by using a large set of images taken in quick succession that show the movement of the vehicle, in other words: a video. Frames extrapolated from a video will show clear and well-defined displacement of the vehicle as we move along the frames. If the system detects the vehicle in each frame, it will be able to quantify the vehicle's displacement. This, coupled with the background that stays in place, provides us with enough information to deduce the speed of the moving vehicle.

Now we have a system that can detect the speeds of vehicles in a video. The traffic police can use this to detect overspeeding vehicles. The system can instantly provide the number plates of these vehicles. If installed at a particular road, it can provide data such as the speed and type of every vehicle that passes through that road. If a sufficient amount of data is gathered, it can be analyzed to determine the average speed of vehicles on the road, most popular vehicle types, times when the road has the most traffic, etc. This analysis can be insightful and useful.

II. BACKGROUND WORK

The field of vehicle detection and classification and speed measurement with the help of computer vision is essential for the effective management of autonomous traffic monitoring systems. Over the years, much research has been done in this field. Several algorithms have been used ranging from Mask RCNN to frame differencing and Gaussian Mixture models to detect and classify vehicles from video surveillance. This section presents a review of previously published paperwork related to vehicle classification and speed estimation.

[1] proposes a novel algorithm titled Video Frame Diminution Technique (VDFT) which is based upon background subtraction method. The selection of an ideal frame of reference is implemented through downsample, which is a frame extraction method that has less computational cost. The speed of the vehicle is estimated by dividing the total number of frames with calibrated speed to that of uncalibrated speed. [2] presents an analytical review of two different vehicle detection and tracking methods namely Faster Region-based Convolutional Neural Network and You Only Look Once (YOLO) algorithm. Moreover, it also describes multiple variants of the two algorithms and the methods based on their respective architectures. Vehicle detection is implemented in [3] using methods such as background subtraction and thresholding along with Mahalanobis distance learning. The speed of the vehicle is estimated by performing division between the distance obtained and the frame rate of the video. [4] implements OpenCV python and video processing techniques such as Median Filtering and Gaussian Filtering for reducing noise within images. Other techniques such as Frame differencing, Thresholding and Contour Extraction, and Bounding Rectangle are used for object detection and speed calculation.

[5] implements Gaussian Mixture Model (GMM) as a background subtraction method for identifying moving vehicles and applies morphological operations such as dilation and erosion for further image processing. The speed of the vehicles is calculated using distance normalisation and centroid calculation. [6] presents a two-step vehicle tracking and detection system using computer vision techniques. Feature extraction is carried out using Histogram of Oriented Gradients while the edge features are estimated using the Haar feature algorithm. A two-stage Faster RCNN detector along with a Simple Online Realtime Tracking (SORT) algorithm is implemented in [7] upon 750 video frames dataset for classification and counting vehicles as well as for calculating their speeds. [8] introduces a video-based speed measurement model that employs an intrusion detection technique and movement pattern vector as an input parameter. To detect the movement of vehicles, a motion estimation approach that uses a polynomial expansion algorithm is implemented. An analytical comparison between different models for object detection such as SSD, YOLOv2, and Faster RCNN is presented in [9]. The use of pre-trained ResNet18 classifier along with transfer learning is used for visual classification. A pre-trained deep learning based YOLOv2 model is implemented for the detection of vehicles in [10]. The tracking of multiple vehicles is performed with the use of open-source Simple Online Real Time Tracking (SORT) algorithm.

III. Dataset Description

We make use of a custom dataset for vehicle detection and classification derived from publicly available Microsoft Common Objects In Context (MS COCO) dataset.

The dataset consists of 5 different classes which are bicycle, car, motorcycle, bus, and truck. We performed various image pre-processing and transformation steps such as image augmentation, image rotation, image flip etc. We have also applied data normalization and scaling on our dataset to help improve the final accuracy of the model. We have converted the detected bounding box format that conforms to that of the object detection algorithm's standard format. Our dataset consists of 19,758 images out of which the training images are 15,807, and the testing as well as validation are 3,951 images.

IV. Model Architecture

4.1 YOLOv5

YOLOv5 is one of the fastest and most accurate state-of-the-art object detection models developed by Ultralytics. It consists of a novel Convolutional Neural Network architecture that has the ability to detect an object in a real-time environment with high precision. This algorithm processes the entire image with a single neural network, then divides it into pieces and calculates bounding boxes and probabilities for each element. The predicted probability is used to weigh these bounding boxes. It produces predictions after only one forward propagation through the neural network. The use of non-max suppression to provide detected items ensures that the detection of an object is carried out only once. The architecture of YOLOv5 is divided into 3 components:

4.1.1 Backbone

Yolov5 integrates a cross-stage partial network (CSPNet) with Darknet, resulting in the CSPDarknet backbone. The main function of the Backbone model is to extract features of prime importance from an input image that is provided to the network. CSPNet eliminates the issue of recurrent gradient information in large-scale backbones by including gradient changes into the feature map, reducing model parameters and FLOPS (floating-point operations per second), ensuring inference speed and accuracy while simultaneously reducing model size. In the detection of vehicles in traffic, speed and accuracy are critical, and the size of the model impacts its inference efficiency on resource-limited edge devices.

4.1.2 Neck

Typically, the Neck is used to construct feature pyramids. When it comes to object scaling, feature pyramids help models generalize successfully. It makes it easier to recognize the same thing in different sizes and scales. Feature pyramids are extremely useful for aiding models in performing well on previously unknown data. Other models, such as FPN, BiFPN, and PANet, utilize feature pyramid methods in various other ways. PANet is used as a neck in YOLOv5. PANet uses a novel feature pyramid network (FPN) topology with an improved bottom-up approach that enhances low-level feature propagation. Simultaneously, adaptive feature pooling, which connects the feature grid to all feature levels, is employed to ensure that meaningful information from each feature level reaches the next subnetwork. PANet enhances the use of precise localization signals in lower layers, which can significantly improve the object’s position accuracy.

4.1.3 Head

The final detection process is primarily handled by the Head of the network. It creates final output feature vectors with class probabilities, object scores, and bounding box using anchor boxes. The head creates three various sizes of feature maps (18 x 18, 36 x 36, 72 x 72) to provide multi-scale prediction, allowing the model to handle tiny, medium, and large objects

4.2 Deepsort

Deepsort is a deep learning-based method for tracking certain objects in video surveillance. Deepsort is used to track vehicles in video surveillance footage. It works by learning patterns from recognized objects in images, which are then paired with temporal data to predict related trajectories of the objects that are of interest. It maps unique IDs to keep track of each object under consideration for statistical analysis. Occlusion, various views, non-stationary cameras, and annotation of training data are all the challenges that Deepsort can conquer. The Kalman filter and the Hungarian algorithm are employed for successful tracking. For improved association, the Kalman filter is applied iteratively, and it may forecast future locations based on the current position. For association and id attribution, the Hungarian method is utilised, which determines if an item in the current frame is the same as one in the previous frame.

4.3 YOLOv3

Using methods like multi-scale prediction and bounding box prediction through logistic regression, YOLOv3 optimized the design even further. While the accuracy of this version improved substantially, it came at the cost of speed, which dropped from 45 fps to 30 fps. Kernels of form 1x1 are applied to feature maps of three different sizes at three distinct points in the network in the convolutional layers. The method generates predictions at three scales, which are achieved by downsampling the image’s dimensions by 32, 16, and 8 pixels, respectively. To minimize the quantity of the data, downsampling is used which reduces spatial resolution while maintaining the same picture representation. Three anchor boundary boxes per layer are used in each scale.

4.4 Model Ensembling (YOLOv5 and YOLOv3)

Ensemble modeling is a process in which several separate models are built to predict a result, either using a variety of modeling techniques or different training data sets. After that, the ensemble model combines the predictions of each base model to provide a single final forecast for the unknown data. The goal of employing ensemble models is to lower the prediction’s generalisation error. When using the ensemble technique, the prediction error of the model lowers as long as the basis models are varied and independent. Despite the fact that the ensemble model has many base models, it functions and performs as a single model.

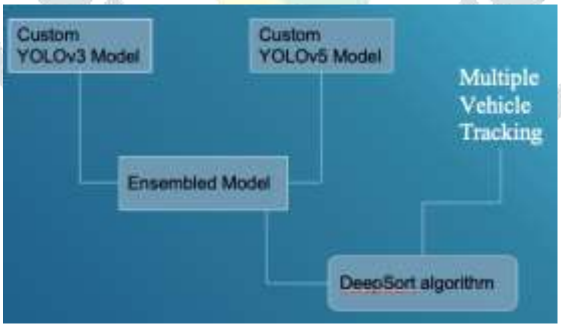

Figure 1: Model Architecture

V. PROPOSED METHODOLGY

Our proposed methodology is divided into 3 different sub sections.

5.1 Vehicle Classification

The role of this module is to take a video from the user that is received from the web portal, and feed it to the trained classification model. The classification model was constructed using the ensembled architecture of YOLOv5 and YOLOv3 object detection algorithms trained on the filtered Microsoft COCO (Common Objects in Context) dataset containing 5 classes: car, truck, motorcycle, bus, and bicycle. Thus, it will detect and then classify the vehicle into one of the above mentioned classes. The module will also process the video by applying the DeepSort algorithm which will detect the movement and trajectory of every vehicle, even if it gets obscured by another vehicle or object. The resultant output is a video with an overlaid bounding box on every detected vehicle in every frame with the respective frame and vehicle id mentioned on top of the bounding box.

5.2 Vehicle Counting

After the classification and detection of vehicles, this module will count the total number of vehicles that pass through the region of interest defined by the user. This is aided further by the use of Deepsort as it assigns a distinct id to every vehicle, which means the number of unique vehicles is equal to the number of unique tracking ids. The purpose of this model is to assess the traffic on the monitored road, as the greater the count, the higher the traffic. It will also provide an estimate to how the traffic changes throughout a day.

5.3 Speed Estimation

Once the vehicles are detected and classified in the given video, this model can begin to calculate their speed. YOLOv3 model is employed for estimation of average speed of the detected vehicles. Estimated speed will be mentioned on the top of bounding boxes of the detected vehicles with their respective ids. A resultant csv file is generated to store details of over-speeding vehicles. The csv file stores information such as vehicle id, speed of the vehicle and if the vehicle was over-speeding or not.

VI. EXPERIMENTAL RESULTS

Our project successfully detects, classifies and tracks different vehicles as well as estimates its speed based on the surveillance video provided. We have trained custom state-of-the-art deep learning models which are YOLOv5 and YOLOv3. The parameter for determining accuracy for object detection models is mean average precision (mAP) which comes out to be 66.90% for YOLOv5 model and 69.20% for YOLOv3 model after training it using Microsoft's COCO (Common Objects in Context) dataset upon 5 different classes of vehicles which are car, motorcycle, bus, truck and bicycle. We have achieved an increased accuracy of 10.1% in our custom YOLOv5 model with respect to the pre-trained YOLOv5 model. Furthermore, we have performed model ensembling between our custom YOLOv3 and YOLOv5 models. For multiple object tracking, we have integrated our ensembled model with DeepSORT architecture that produces high precision and accuracy. The speed of the vehicles is estimated using the YOLOv3 model.

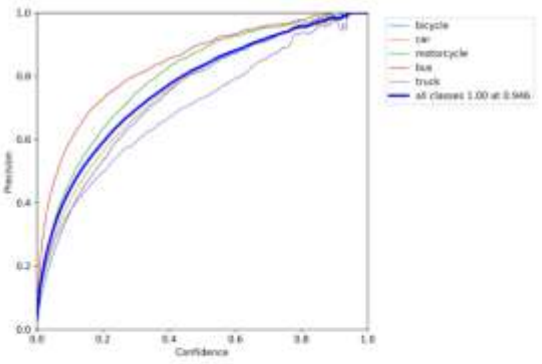

Figure 2: P Curve for YOLOv5

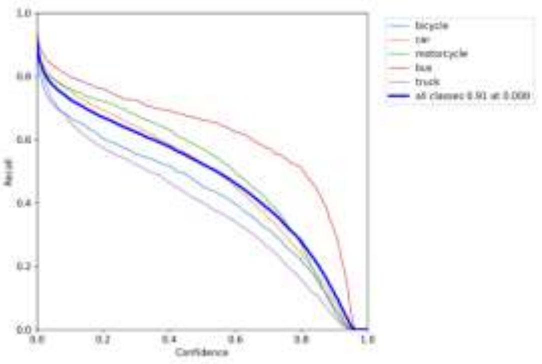

Figure 3: R Curve for YOLOv5

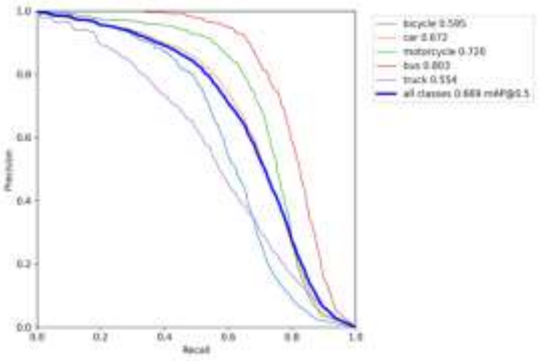

Figure 4: PR Curve for YOLOv5

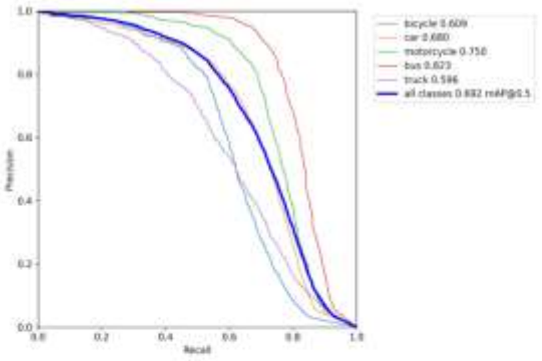

Figure 5: PR Curve for YOLOv3

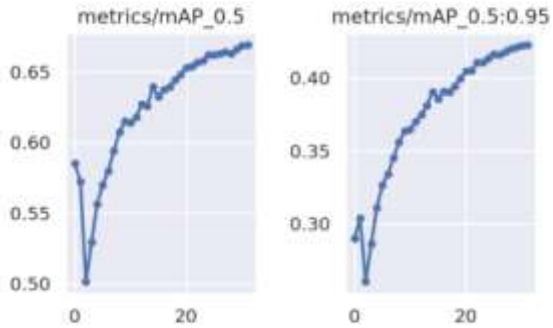

Figure 6: mAP Curve for YOLOv5

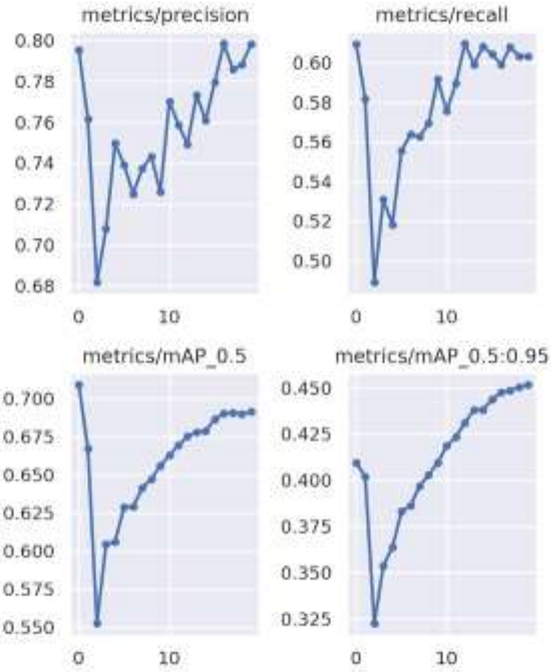

Figure 7: Other metrics for YOLOv3

VII. Conclusion

Vehicle speed detection is a complex task with various real-life difficulties. Online traffic analysis via our platform eliminates the need for human labor needed for automated vision-based tasks that increase accuracy of models via model ensembling. Traffic analysis is important for monitoring the activity of vehicles, detecting rash driving and vehicular movement on the roads or highways. The generated analysis report contains information of over-speeding vehicles. By entering the required surveillance video into the system, the user can obtain the analysis report. In conclusion, this solution has been prepared to provide a framework to the research community that can be used with a larger data set and can be combined with other resolution video data. With further tuning on large and diverse data sets, the algorithm may be used in real-time to help the traffic department to detect and manage overspeeding.

References

[1] Pillai, V. J., Kumar, K. P., Prathap, B. R., & Chandra, S. (2021). Fixed Angle Video Frame Diminution Technique for Vehicle Speed Detection. Annals of the Romanian Society for Cell Biology, 3204-3210.

[2] Maity, M., Banerjee, S., & Chaudhuri, S. S. (2021). Faster R-CNN and YOLO based Vehicle detection: A Survey. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1442-1447). IEEE.

[3] Chandorkar, M., Pednekar, S., & Bojewar, S. (2021). Vehicle Detection and Speed Tracking. International Journal of Engineering Research & Technology (IJERT) Volume 10, Issue 05.

[4] Berna, S. J., Swathi, S., & Devi, C. Y. (2020). Distance and Speed Estimation of Moving Object using Video Processing. International Journal for Research in Applied Science and Engineering Technology (IJRASET) Volume 8, Issue 5.

[5] Agrawal, S. C., & Tripathi, R., K. (2020). An Image Processing Based Method For Vehicle Speed Estimation. International Journal of Scientific & Technology Research (IJSTR) Volume 9 Issue 4.

[6] Gaikwad, M., Joshi M., Desai R., & Patil, B. (2021). Vehicle Detection using Haar and HOG Feature Extraction Algorithms and SVM with Speed Estimation. International Journal for Research in Applied Science and Engineering Technology (IJRASET) Volume 9 Issue 5.

[7] Grents, A., Varkentin, V., & Goryaev, N. (2020). Determining vehicle speed based on video using convolutional neural network. Transportation Research Procedia, 50, 192-200.

[8] Javadi, S., Dahl, M., & Pettersson, M. I. (2019). Vehicle speed measurement model for video-based systems. Computers & Electrical Engineering, 76, 238- 248.

[9] Liu, C., Huynh, D. Q., Sun, Y., Reynolds, M., & Atkinson, S. (2020). A vision- based pipeline for vehicle counting, speed estimation, and classification. IEEE Transactions on Intelligent Transportation Systems.

[10] Bell, D., Xiao, W., & James, P. (2020). Accurate vehicle speed estimation from monocular camera footage. In XXIV ISPRS Congress. Newcastle University.