WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Video Based Vehicle Detection and Tracking using Image Processing

Suruchi Kumari, Deepti Agrawal

Department of Electronics and Communication Engineering, School of Research and Technology
Peoples University Bhopal, Madhya Pradesh, INDIA
geetasuruchi@gmail.com, er.deeptiagrawal@gmail.com

A B S T R A C T

In the subject of highway management, intelligent vehicle system and traffic management are becoming increasingly crucial in today's time. As various vehicles are seen on the street daily, the number of these vehicles is increasing at a high rate. With this huge amount of vehicles, detection and tracking, become a difficult job especially in developing and underdeveloped countries. With this thought, detecting and tracking vehicles from video frames is discussed and addressed in this work. In the realm of object detecting and tracking, the deep learning method has been widely applied. This work proposes video-based vehicle detection and tracking. A new high definition highway vehicle data set containing more than 10,000 images extracted from videos with proper annotation is made from the different area, which provides a complete data foundation for vehicle detection and tracking based on deep learning on Python platform. For making this data, different types of image processing methods had been used. For detection purposes, the YOLO v5 model, which is the latest version of the YOLO model, is being used; also, the MASK R-CNN model and SSD (Single Shot Detection) have been used for detection purposes. For tracking, we have used the YOLO v5 model, Deep SORT framework and GOTURN method. After detection and tracking, vehicle count and speed estimation are done. For vehicle counting and speed estimation, the YOLO v5 model is used. This information will aid in determining the priority and maximum users of a route and designing traffic patterns that will benefit the most people. Several highway surveillance videos based on different scenes are used to verify the proposed method.

Keywords:Video based detection, Image Processing, SSD.

1. Introduction

Intelligent transportation systems are crucial in today's climate for traffic management in order to establish an efficient and dependable transportation system. The precise identification and monitoring of vehicles is one use of the intelligent transportation system. Now, vehicle information collecting technologies mostly comprise loop coil detection, infrared detection, and intelligent video surveillance detection. The loop coil detection operation is steady, the detection accuracy is excellent, and the traffic information can be tallied among them. It is simple to install and configure, and it is commonly utilised in locations such as road tollbooths and parking lots. To measure vehicle speed, infrared detection mostly use light-emitting diodes with high detection sensitivity. However, it is easily impacted by environmental factors such as temperature, humidity, and so on, resulting in low detection accuracy and resilience. Intelligent video surveillance and detection has become more crucial in traffic information collecting as technologies such as image recognition and computer vision have advanced. Intelligent video surveillance detection employs a camera installed at a traffic intersection to do target analysis on the camera-monitoring region in order to collect unstructured information about the target in the video. Video traffic monitoring offers a plethora of information and serves as an important data source for intelligent traffic monitoring systems. Fixed cameras, on the other hand, have restricted data resources. Drone technology has begun to be widely employed in traffic monitoring, with rich data kinds and rapid data collecting, as it has developed. The problem of the research is determining how to recognise the vehicle among the vast amounts of data.

* *Corresponding author.* Tel.:+919304214651;
E-mail address: geetasuruchi@gmail.com

2. Image Processing

Image processing seeks to convert an image into digital form and apply some procedure to it in order to obtain a better image or extract useful information from it. A technology is being developed to transform images into digital form and execute various operations on them in order to obtain particular models or extract important information from them. This technique takes as input a video segment or a picture, such as a photograph. The output matches to the intended or attention-grabbing portion of the image. Digital image processing techniques aid in the alteration of digital pictures with computers. Pre-processing, augmentation and presentation, and information extraction are the three main processes that all sorts of data must go through when employing digital techniques. (Source: Figure 1).

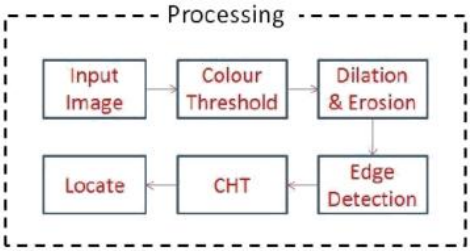

Figure 1 Basic Structure of Image Processing

Threshold

Thresholding is the most basic approach of picture segmentation in digital image processing. Thresholding may be used to produce binary pictures from grayscale photographs.

Dilation

Dilation adds pixels to the edges of objects in a picture, whereas erosion removes pixels from the edges of things. The state of every given pixel in the output picture is defined by applying a rule to the relevant pixel and its neighbours in the input image during the dilation and erosion procedures. The procedure is defined as a dilation or erosion by the rule used to process the pixels. The value of the output pixel in a dilation operation is the largest value of all pixels in the vicinity. A pixel in a binary picture is set to 1 if any of its neighbours contain the value 1. The value of the output pixel in an erosion process is the least value of all pixels in the vicinity. A pixel in a binary picture is set to 0 if any of its neighbours contain the value 0.

Edge Detection

Edge Detection is a technique for segmenting a picture into discontinuous parts. It is a popular approach in digital image processing applications such as pattern recognition, picture morphology, and feature extraction. Edge detection allows users to examine picture features for substantial changes in grey level. This texture marks the end of one section of the picture and the start of another. It decreases the quantity of data in a picture while preserving its structural features.

CHT

The circle Hough Transform (CHT) is a fundamental feature extraction approach in digital image processing that is used to find circles in defective pictures.

Locate

Identifying the position of one or more items in a picture and creating a bounding box around their extent is referred to as object localization. Object detection integrates these two objectives by locating and classifying one or more items in a picture.

2.1 Proposed Methodology

We attempted to experiment with identifying automobiles from video using our proprietary dataset. As a flow chart diagram, we offer our experimental methods below (Figure 2). Each phase will be described in full later.

The method of recognizing and tracking moving objects in a video is known as object detection and tracking. A person, animal, or vehicle can all be considered objects. Using efficient methods and algorithms, we are able to recognise and track all types of cars in a given video. Therefore, in our suggested technique, we first created a customized dataset by transforming films into frames with appropriate annotation. Then we divided our customized dataset in half, keeping 80 percent for training and 20 percent for testing. The detecting module extracts and identifies the target item, which is subsequently sent to the learning and tracking modules, respectively. We utilised YoloV5, SSD, and Mask RCNN to detect our automobiles. To construct the trajectory for tracking, the locations of the object must be found in two consecutive frames. We utilised Deep Sort and GOTURN for tracking, and we assessed the speed of each car as well as tallied the number of vehicles. We manually evaluated these values for the existing route in order to compute pixels per metre (ppm). We took a video frame and estimated the road width in pixels. Then we converted pixels to metres and obtained the results using the speed formula (speed = dmeters X fps X 3.6). We created a counter for counting and split the route into polygons to designate lanes. Any class object that falls within the polygon region gets boxed, and the counter is incremented. As a result, we are counting, tracking, and calculating the speeds of the cars.

Dataset

We have made our own custom dataset consists of local vehicles information of our country.

Data Collection

Because we are going to recognise and track local automobiles, the dataset must be solid and have quality characteristics. As a result, we gathered data by taking video photos from various perspectives that included accurate vehicle information.

Data Preprocessing

Took films of traffic on several highways and then produced graphics from those videos. Images are chosen with automobiles in the frame, in plain view, and without blurry things in mind. Images are scaled to 64 px X 64 px. And the size is lowered to about 40kb for training efficiency. Images are sorted in numerical order to keep track of them.

Data Annotation

More than 10000 frames have been annotated with vehicle bounding boxes indicated. Our dataset includes films with a wide range of size, position, lighting, occlusion, and background clutter. We have introduced a total of five classes, which include:Bus, Bike, Cycle, Car, Truck and Van.

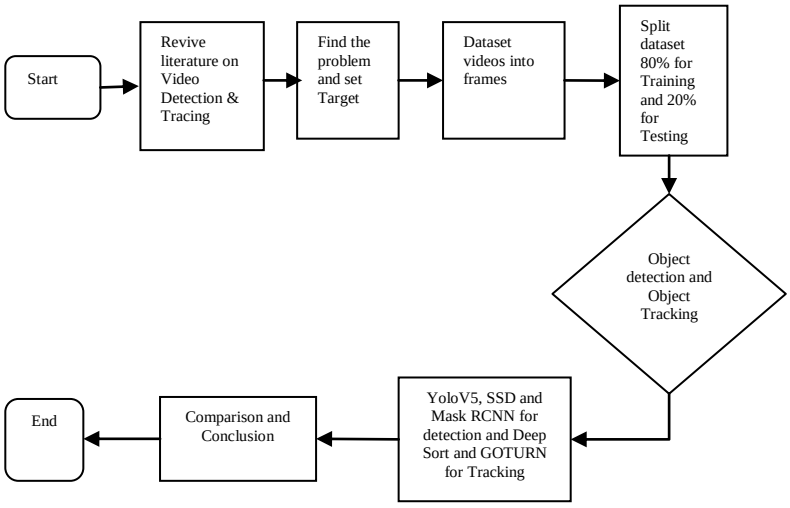

Figure 2Methodology Diagram

2.2 Prediction Rate of Detecting Vehicles

When it comes to tiny objects, SSD performs significantly worse than Mask R-CNN. The fundamental reason for this disadvantage is because higher resolution layers in SSD are in charge of recognizing little items. Mask R-CNN accuracy comes at the expense of temporal complexity. It is substantially slower than YOLO, for example. YOLO v5 is one of the greatest object detecting system changes models ever created. Graphs of three models reveal that the YOLO V5 has the best overall performance and is the most efficient.

2.3 Prediction Rate of Tracking Vehicles

To track cars, we must establish the target's starting state by drawing a bounding box around it, estimate the target position, and learn the target's visual appearance, motion estimates, and actual location. For tracking, we utilised GOTURN. It just tracks one thing, but it tracks it well. To monitor many cars in our dataset, we utilised YoloV5 and the Deep Sort method. Furthermore, the Deep SORT tracking cannot track the object if the YOLO cannot identify any bounding box of this item, resulting in the object tracking about the identity swaps degrading. Yolo V5, on the other hand, can identify and track things. In comparison, the YoloV5 tracker outperformed the present real-time Deep SORT tracking algorithm in terms of tracking accuracy.

3. Algorithm Design

Vehicle Classifier Design Ideas to achieve vehicle detection from the standpoint of deep learning, develop a vehicle detector using the CNN approach in deep learning. The vehicle detector is implemented by determining whether or not the item in each detection frame is a vehicle. As a result, in order to achieve the two-class classification of vehicles and non-cars, a vehicle classifier with better CNN must be designed. Figure 3 depicts the training and testing of the classifier model. Perform a pre-processing procedure on the training samples first. The training samples and labels are then sent into CNN for training, yielding an improved CNN model. The test samples are then pre-processed, and the model's prediction results are achieved by enhancing the CNN model. The prediction result is compared to the test sample label to produce the final test result.

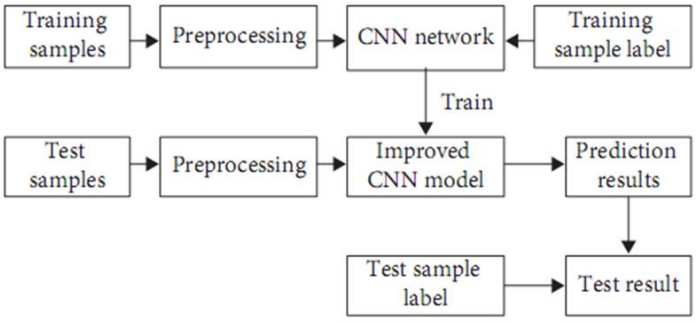

Figure **Error! No text of specified style in document.** Process of training and testing CNN classifier

3.1 Dataset Preprocessing

Sample Collection

The trained classifier's input picture size is set to 50 50, which is also the size of the vehicle detection frame in the real video. Cropping and scaling techniques are used to obtain original samples that fulfil the requirements for photos whose sample size does not match the requirements. Because there is no vehicle sample collection that contains various models and angles. 2503 positive samples and 2,698 negative samples are collected from several videos and photos to generate a pretty full vehicle classification data set. The images in the positive sample set depict automobiles of various brands and models, as well as diverse camera perspectives. The light conditions and backdrop of each sample are likewise diverse, ensuring a wide range of positive results. It is sufficient to demonstrate the suggested classifier's flexibility if it can recognise these sorts of cars with varied backgrounds. The negative sample collection contains a variety of backdrop sceneries devoid of automobiles that may exist along the road. There might be road signs, tree shadows, people, trees, and so on, all of which are sufficiently diversified.

Data Preprocessing

The original data set is preprocessed in the HSV space to improve the classifier's flexibility to diverse lighting situations especially in the shadow. The V value in the HSV colour space denotes the brightness, which is the color's brightness, and it is also given as a percentage. Its value varies from 0% to 100%, with 0% being entirely dark and 100% being fully brilliant. As a result, the brightness can roughly represent the brightness of the complete picture, and the brightness of the scene can be altered using the brightness shift operation. Increase the average V value of each original photo by 20%, 30%, 40%, 50%, and 60%. (because it is too dark at 20 percent , and the scene is too bright at 70 percent). While maintaining the H and S values constant, the new V value for each point is:

$$V_{\text{new}}(a,b) = \frac{V_{\text{old}}(a,b) \times \overline{V}_{\text{new}}}{\overline{V}_{\text{old}}}, \qquad \text{Eqn. 1}$$

where V new (a, b), V old (a, b), V new, V old represent the new and old lightness values, the new and old lightness mean values, and the old lightness mean value, respectively. The above procedure yields a sample set including a wide range of brightness values. The sample collection now contains 15,426 positive samples and 17,071 negative samples. Before you begin processing, transform the picture from RGB (Red-Green-Blue) space to HSV space. Following processing, all images must be converted back to RGB space. Following that, all of the samples processed before are greyed out. The components of the R, G, and B channels of the pixel at location (a, b) are denoted as R (a, b), G (a, b), and B (a, b), respectively. The grey value of the associated spot is then:

$$G(a,b) = 0.2989 \times R(a,b) + 0.5870 \times G(a,b) + 0.1140 \times B(a,b). \qquad \text{Eqn. 2}$$

The grayscale value range of the grayscale picture after grayscale is [0, 255]. Finally, to retrieve the input layer sample data, execute a simple normalisation operation on the grayscale picture data:

$$I(a,b) = \frac{G(a,b)}{255}. \qquad \text{Eqn. 3}$$

4. Result Analysis

This section discusses simulation findings and observed performance factors like as accuracy, precision, and recall. It also emphasises the confusion matrices of various datasets and the techniques' convolution layers.

Experiment and Analysis

The hardware and operating environment of the experiment are shown in Table 1 and Table 2 respectively.

Table 1Hardware Requirement

Processor	**Intel i7, 8th Gen quad core**
Clock Speed	**1.8 GHz**
RAM	**16 GB**
Storage	**500 GB SSD**
GPU	**Nvidia MX**

Table 2 Software Requirements

Distribution	**Anaconda Navigator**
API	**Keras**
Library	**Tensor Flow, OpenCV**
Packages	**Matplotlib, numpy, pandas, scikit Learn**
Language	**Python**
IDE	**Spyder, Jupyter Notebook**
GPU Architecture	**CUDA**
Applications	**Label Img, Tensor Board**

4.1 Single Object Detection

CNN is designed for single object detection. It includes the settings used in each step, the layer progression, and the output image size of each layer. Each layer deconstructs the picture matrix and performs an operation on the image. The output picture size of multiple layers varies owing to manipulations performed by each layer, such as when the output image size is 2828, which is subsequently reduced to 1414 by the max-pooling layer, which selects the highest valued pixel from the surrounding pixels. The second max-pooling layer then drops it to 77. This pixel is then flattened into 7764 vectors of size 3136. Following layers decrease this vector to a smaller size, and the final computation parameters are presented.

Training this model yielded an 82 percent training accuracy. The loss and the precision are inversely proportional. As the number of epochs grows, so does the learning rate, and therefore the loss. The model trains itself at each epoch, and the weights of the convolution networks are updated to a more accurate value.

The CNN is successfully able to classify the given object as truck and car with an accuracy of 75.68% and 84.409% respectively as shown in Figure 4.

Figure 4 Sample Simulation Results of Images

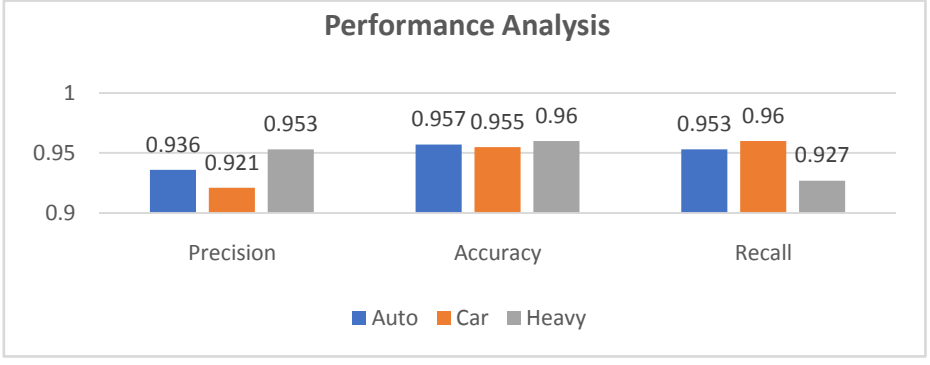

Figure 5 Performance Analysis of Image

4.2 Multiple Object Detection

Figure 6 YOLOv5 Results for COCO Dataset

The photographs gathered from have been provided as a test set. The collection consists of three types of photos: day, evening, and NIR images (Near Infrared. Figure 6 illustrates that the system can recognised objects of any size and photos collected from different camera angles and distances. This property is due to the FPN used in YOLOv5. The accuracy is great since the items to be recognised are notobstructing one another, and the recall is 0.8333 because the FN value is 1 becausethe auto in the image is detected but incorrectly labelled as a truck. The recall value suffers because of misclassification. Because the accuracy is great,the mAP value for the given class and picture is 100 percent.

4.3 Multiple Object Tracking

YOLOV5 and deep learning approaches were used to train the vehicle tracker on surveillance footage. The vehicle tracking process was completed successfully by testing a trained vehicle detector on test data set video. The algorithm split the movie into frames at a rate of 30 frames per second and detected objects in the first frame. The recognised picture was tracked using its centroid position in the later frames.

Table 3Comparison Result of models using custom data set

Models	Precision	Recall
SSD	**70.1**	**32.9**
Mask R-CCN	**37.9**	**54.2**
YOLOV5	**72.4**	**70.2**

Figure 7Comparison Result of models

5. Conclusion

The use of artificial intelligence to tackle computer vision challenges surpassed traditional image processing methodologies. For day photos, the CNN model trained on an on-road vehicle dataset for single object recognition obtained validation accuracy of 95.7 The large amount of data on which it is

trained from each class accounts for the excellent validation accuracy. Images' performance data are recorded. For the KITTI and COCO datasets, multiple object detection is accomplished using YOLOv5.

On the considered classes of photos, performance metrics for YOLOv5 are tabulated. The bigger the class's precession value, the greater the mAP value. The mAP value is determined by the picture used in the computation. A detection and tracking IoU of 0.5 is excellent. By raising genuine positive values, mAP values can be improved. The outcome of performance measures is entirely reliant on the picture data set utilised. Based on the region of interest, further items are discovered in the movie. The performance parameters examined include vehicle speed and color, vehicle type, vehicle movement direction, and the number of vehicles in ROI. YOLOv5 and OpenCV are used to implement multiple object tracking in traffic surveillance footage. On different frames of a movie, many objects are recognised and tracked.

REFERENCES

A. Ambardekar, M. Nicolescu, and G. Bebis, "Efficient vehicle tracking and classification for an automated traffic surveillance system," Signal and Image Processing, 2008, pp. 1-6.

A. Appathurai, R. Sundarasekar, and C. Raja, "An efficient optimal neural network-based moving vehicle detection in traffic video surveillance system," Circuits, Systems and Signal Processing, 2020, vol. 39, no. 2, pp. 734-756.

A. I. B. Parico and T. Ahamed, "Real time pear fruit detection and counting using yolov4 models and deep sort," Sensors, 2021, vol. 21, no. 14, p. 4803.

B. Yca, B. Bm, and C. Hong, "Part alignment network for vehicle re-identification -Science Direct," Neuro computing, 2020, vol. 418, no. 5, pp. 114-125.

D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in European conference on computer vision, 2016, pp. 749-765.

D. Sudha and J. Priyadarshini, "An intelligent multiple vehicle detection and tracking using modified vibe algorithm and deep learning algorithm," Soft Computing, 2020, vol. 24, no. 21, pp. 1-13.

E. Bochinski, T. Senst, and T. Sikora, "Extending iou based multi-object tracking by visual information," in 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2018, pp. 1-6.

E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2017, pp. 1-6.

H. Tao and X. Lu, "Automatic smoky vehicle detection from traffic surveillance video based on vehicle rear detection and multi-feature fusion," IET Intelligent Transport Systems, 2019, vol. 13, no. 2, pp. 252-259.

J. Athanesious, V. Srinivasan, and V. Vijayakumar, "Detecting abnormal events in traffic video surveillance using super orientation optical flow feature," IET Image Processing, 2020, vol. 14, no. 9, pp. 1881-1891.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, realtime object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961-2969.

M. Bugeja, A. Dingli, and M. Attard, "Comparison of vehicle detection techniques applied to IP camera video feeds for use in intelligent transport systems," Transportation Research Procedia, 2020, vol. 45, no. 5, pp. 971-978.

M. Sankaranarayanan, C. Mala, and S. Mathew, "Pre-processing framework with virtual mono-layer sequence of boxes for video based vehicle detection applications," Multimedia Tools and Applications, 2020, vol. 5, no. 6, pp. 1-28.

M.-R. Lee and D.-T. Lin, "Vehicle counting based on a stereo vision depth maps for parking management," Multimedia Tools and Applications, 2019, vol. 78, no. 6, pp. 6827-6846.

Mohana and HV Ravish Aradhya, " Object Detection and Tracking using Deep Learning and Artificial Intelligence for Video Surveillance Applications," International Journal of Advanced Computer Science and Applications, vol. 10 no.12, 2019, pp.517-530.

P. Martinez and M. Barczyk, "Implementation and optimization of the cascade classifier algorithm for UAV detection and tracking," Journal of Unmanned Vehicle Systems, 2019, vol. 7, no. 4, pp. 296-311.

P. Priyadharshini and P. Karthikeyan, "Vehicle data aggregation from highway video of madurai city using convolution neural network," Procedia Computer Science, 2020, vol. 171, no. 4, pp. 1642-1650.

Q. Zhang, H. Sun, X. Wu, and H. Zhong, "Edge video analytics for public safety: a review," Proceedings of the IEEE, 2019, vol. 107, no. 8, pp. 1675-1696.

R. A. Hadi, G. Sulong, and L. E. George, "Vehicle detection and tracking techniques: a concise review," 2014.

R. Feng, C. Fan, and Z. Li, "Mixed road user trajectory extraction from moving aerial videos based on convolution neural network detection," IEEE Access, 2020, vol. 8, no. 4, pp. 43508-43519.

R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," IEEE transactions on pattern analysis and machine intelligence, 2015, vol. 38, no. 1, pp. 142-158.

S. Houben, M. Neuhausen, and M. Michael, "Park marking based vehicle self-localization with a fisheye top view system," Journal of Real-Time Image Processing, 2019, vol. 16, no. 2, pp. 289-304.

S. S. Esfahlani, "Mixed reality and remote sensing application of unmanned aerial vehicle in fire and smoke detection," Journal of Industrial Information Integration, 2019, vol. 15, no. 3, pp. 42-49.

Shaba Irram, Sheikh Fahad Ahmad, "Research on Object Detection in Video Streaming Using Deep Learning," International Journal of Computational Engineering Research, vol. 09, July-2019, pp. 34-43.

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

W. Zhan, C. Sun, M. Wang, J. She, Y. Zhang, Z. Zhang, and Y. Sun, "An improved yolov5 real-time detection method for small objects captured by uav," Soft Computing, 2021, pp. 1-13.

X. Peng, Y. L. Murphey, and R. Liu, "Driving maneuver early detection via sequence learning from vehicle signals and video images," Pattern Recognition, 2020, vol. 103, no. 4, pp. 107276-107286.

Xiangqian Wang, "Vehicle Image Detection Method Using Deep Learning in UAV Video," Hindawi Computational Intelligence and Neuroscience, vol. 2022, pp. 1-10. February-2022.

Y. Cao and X. Lu, "Learning spatial-temporal representation for smoke vehicle detection," Multimedia Tools and Applications, 2019, vol. 78, no. 6, pp. 27871-27889.

Y. Huang and H. Zhang, "A safety vehicle detection mechanism based on yolov5," in 2021 IEEE 6th International Conference on Smart Cloud (Smart Cloud), IEEE, 2021, pp. 1-6.

Z. Liu, D. Lu, W. Qian., "A method for restraining gyroscope drift using horizon detection in infrared video," Infrared Physics & Technology, 2019, vol. 101, no. 3, pp. 1-12.