Deep Learning Model for Vehicle Taillight Detection and Recognization in Autonomous Driving

Gajula Mounika [1], Vasundra S [2]

[1] *PG Scholar, Department of CSE, JNTUACEA, Anantapuramu, India*
mounikagajula543@gmail.com
[2] *Professor, Department of CSE, JNTUACEA, Anantapuramu, India*
vasundras.cse@jntua.ac.in

***Abstract*— Automated cars are a technical advancement in the automobile industry. Automated vehicle detection can be employed as a part of forward collision avoidance and mitigation systems. Vehicles in front are typically detectable by their taillights when driving in low light. Real-time detection and identification of taillights can aid in preventing traffic incidents that may result from a driver's disregard for them. In this paper we have proposed YoloV5 model, which detect car taillight in autonomous driving. The results showed that the model perform well on a variety of criteria with 50 training iterations, the model achieved accuracy of 92.36%.**

Keywords—Autonomous Driving, YOLO, Taillight, Deep Learning (DL), Vehicle Detection.

I. Introduction

Automated cars are a technical advancement in the automobile industry. Even while autonomous cars are safer for people, they are also the most expensive. A computer-driven vehicle that can steer itself, get familiar with its environment, make judgments, and function entirely independently is referred to as an autonomous vehicle. Over the past few decades, various industrial businesses and research organizations have given autonomous driving a lot of attention, leading to significant improvements. A fundamental prerequisite for autonomous cars is the ability to follow other vehicles safely and behave appropriately to avoid mishaps and rear-end collisions. Recognizing the signal from the rear lights helps a driver comprehend the actions and intentions of the car in front of them. Despite the enormous success, various high-level tasks, including as the forecasting of other road users' intentions, human-vehicle interactions, and the connection between self driving cars and hand driving cars were always up for debate, where a crucial necessity is to precisely grasp the intended movement of other vehicles.

In actuality, the majority of traffic-related communications are carried out via signals from moving cars' lights, particularly their taillights. As a result, real-time detection and identification of taillights can aid in preventing traffic incidents that may result from a driver's disregard for them.

Three categories of vehicle identification using visual sensors may be made: functionality, ML-based, and DL-based. The term "feature-based vehicle identification approach" refers to identifying a vehicle by using its fixed properties, including such symmetry, lights, and shadows. This approach has a high processing speed, despite the fact that it has weak algorithm resilience and is heavily impacted by the environment. The term "ML-based vehicle identification technique" encompasses the collection of features from a large number of samples and the training of the classifier using classification algorithms. However, the detecting frame regression is insufficiently precise due to the high computational cost and temporal complexity of the thresholding approach utilized in its prediction. Due to its quick computation speed, good accuracy, and capacity to simultaneously detect numerous species, deep learning is now the most popular approach in the field of machine vision. Deep learning-based target identification technology is being utilized more and more often as DL and GPU technologies progress [1]. A high-performance server is the only place where some regularly used detection network information can be realized in real-time due to their size. Therefore, one of the difficulties of current research is to incorporate vehicle detection in sensing devices.

Artificial neural networks, also known as deep learning, are a branch of machine learning that take their inspiration from the structure and operation of the human brain. Given their outstanding performance in this domain, convolutional neural networks (CNN), which were utilised in this work, are the most current deep learning technologies. Since the aforementioned techniques may be used to both characterize and identify the particular component in an image, both classification and target identification were utilized.

The YOLO series is one of the best object detection models (You Only Look Once). In contrast to previous area proposal-based algorithms, it splits the input picture into a S x S grid and anticipates the and frequencies boxes for an item whose centre falls within a grid cell. Our chosen trained model accurately predicted almost all orientations, angles, and aspects, with a mean average accuracy of 92.36%.

II. Related Work

J.G. Wang et al. [2] introduced Appearance-Based Brake-Lights Recognition Using Deep Learning and Vehicle Detection. A quick vehicle identification technique generates a database of rear vehicles, which is used to train an eight-layer neural network convolutional called the BVLC AlexNet model. The adoption of a two-stage technique shields picture noise-induced brake-light identification issues from taillight pairing issues. Whether the target car and the autonomous car are in the same lane or not, the system can detect brake lights.

F. I. Vancea et al. [3] examined two alternative approaches, a color-based thresholding approach and a deep learning approach based on convolutional neural networks for recognizing vehicle taillights. For tracing the taillights, the deep learning technique performed better when combined with a Kalman filter.

H.T.Chen et al. [4] created a vision-based method that uses photos from a driving video clip as its input for daytime brake light identification. This method involved extracting the previous cars and verifying their taillight symmetry. Then, radial symmetry and brightness characteristics were combined to find the brake light. Retrieving and recovering the missing targets also involved using a refining procedure with temporal information.

M. Casares et al. [5] demonstrated a reliable and compact system for detecting the front vehicle's turn signals & brake lights, including both nighttime and throughout the day. here tcombines the rear-light codebook with Kalman filter tracking. Videos shot during the day and night were used for experiments. When required, correction methods are applied to update the undiscovered lights' codebook and the internal state of the Kalman filter. These strategies greatly raised the rates of detection and categorization.

G. Zhong et al. [6] developed a learning-based technique to forecast the brake light condition. To be more precise, they used an object detector and tracker to locate the automobiles first, and then they recognized the zones that contained the brake lights by utilising colour information and morphological operators. Then, for the purpose of final recognition, an SVM classifier was built using both colour and hierarchical characteristics.

S. Nedevschi et al. [7] developed a convolutional neural network model to find the vehicles' relative orientation and taillight. A Faster RCNN in particular was used to detect automobiles and categorise their orientations, while a sub-network was used to segment the pixels in the taillight and determine its signal condition. Recently, a number of end-to-end deep learning models have been used to the detection of vehicle lights.

S. Velipasalar et al. [8] proposed a reliable and computationally efficient approach for a vehicle vision system that can identify and track car taillights as well as forecast turn and braking signals. In particular, area tracking and intensity tracking were used to detect signals.

D. Nava et al. [9] suggested a real-time method for detecting and classifying brake lights, in which a lane detection algorithm as well as a YOLO model were employed to detect preceding vehicles , and SVM was used to distinguish various states of brake lights. The second category of methods uses distinct deep learning sub-networks without an end-to-end approach for vehicle detection and taillight detection.

Q. Li et al. [10] develops an improved YOLOv3-tiny model-based real-time vehicle taillight recognition technique. First, the number of prediction layers is increased by using the feature fusion technique, and the detection of small objects is aided by the fusion of feature maps across various depths. Second, a spatial pyramid pooling module is used to enrich deep features within a layer using data from various receptive fields. The goal of these two methods is to create multi-scale characteristics that may be used to locate things of various sizes. This variant offers YOLOv3-tiny considerable upgrades at a relatively low additional computational cost. We also create a dataset for the detection of taillights using photos that have been meticulously annotated with bounding boxes for autos and taillights.

III. PROPOSED WORK

This section outlines the dataset, proposed approach and model training for vehicle taillight detection and reorganization in autonomous driving.

A. *Dataset*

This study made use of a taillight dataset of 4032 images. It includes images of the taillight and car classes. The dataset is made up of two sets of labels and images. YouTube CCTV videos, GitHub repositories, and publicly accessible websites are used to collect the data required to detect taillight. Remove any noisy data from the dataset using image restoration, then resize the images to fit the YOLO format. The dataset has training and testing sets. This dataset is used for testing 20% of the time and for training 80% of the time. The YOLOV5 dataset is made up of labels and images. The bounding box parameters are included in these labels. A yaml file was present in YOLOV5. The yaml file is given dataset information, such as the number of classes, class names, and dataset path.

B. *You Only Look Once (YOLO)*

This project aims to distinguish between a car and a taillight during autonomous driving In this study, an object detection model is being used. For object detection, the model comprises two categorics. The first is a recurrent convolutional neural network (RONN), and the second is a YOLO (You only Look Once) series. The YOLO series is best suited for those who can leam rapidly and accurately.
This YOLO series has five variations: versions 1, 2, 3, and 4. YOLOVS is currently the most recent version The first of the YOLO models to be constructed on the PyTorch framework rather than Darknet is YOLOvs, which is significantly more transportable and user-friendly than prior versions YOLOVS comes in four different model variations. They go by the abbreviations YOLOVSs (Small),

YOLOVSin (Medium), YOLOVSI (Large), and YOLOVSx (Extra-large). In all ofthem, YOLOVSs are easy to apply. Ir is used in this work because YOLOVSs requires minimal storage space and offers good accuracy. For object detection, YOLOVSs uses the bounding box regression method (11]. The image's bounding boxes each have the following characteristics: If an object is present in each grid, its bounding box centre (Bx By), Class (C), such as a car or taillight, width (Bw), height (Bb) and PC indicate it. If an object is present, its value is 1, otherwise it is 0

The YOLO principle is

$$Y= (C, Bw, Bh, Bx, By, Pc)$$

Model Head, Model Neck, and Model Backbone are the three elements that make up the architecture of YOLOV5, as depicted in Figure 1. The basic objective of Model Backbone is to extract crucial information from input data. to take a source image and pull out important details from it. In YOLO v5, the framework for extracting specific information from the images is called Cross Stage Partial Networks (CSPDarknet). Model Neck's primary goal is to build feature pyramids. In general, models can scale images effectively based on feature pyramids. It is advantageous to be able to recognise the same thing across a range of scales and dimensions. Feature pyramid models operate well on unseen data. Additional models are utilised for other feature pyramid approaches, including the Feature Pyramid Network (FPN), the BiFPN, and the Path Aggregation Network (PANet). The model Head is largely used for the final detecting stage. The final output vectors were produced using bounding boxes, objectness scores, and confidence scores using the bounding box method to apply to the features.

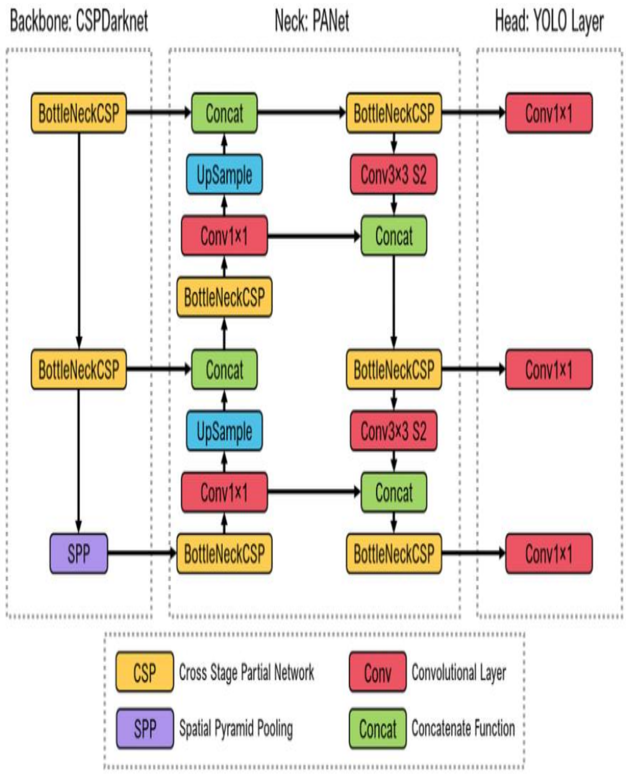

Fig 1. Architecture of YOLOV5.

C. Model Training

The YOLOV5 method detects objects using Pytorch and runs the taillight dataset using Google collaboratory. The YOLOV5 repository was acquired for cloning from the YOLOv5 GitHub library. This can be found at https://github.com/ultralytics/yolov5. To access the dataset, mount a Google Drive into a Google Colab Notebook. The next step is to construct a yaml file that stores, the number of classes, and the class names, location of the dataset as shown below.

train:/content/drive/MyDrive/taillight_dataset/train/images
test: /content/drive/MyDrive/taillight_dataset/test/images

class names:

0: car

1: taillight

The model will be trained for 50 epochs with a batch size of 32 and a learning rate of 0.001. This is the saved model path after training: runs/train/exp/weights/best.pt. After the model has been trained, it should be evaluated for vehicle taillight detection. Finally, the system recognizes the vehicles by their taillights. Predicted images following automobile and taillight detection are saved in the specified path (runs/detect/exp).

IV. EXPERIMENTAL RESULTS & DISCUSSION

All of the experiments in this paper are performed out using 4GB RAM, an Intel Core i5, 7th generation CPU, and a Google Collaborator GPU with 4GB memory. The YOLOV5 system was trained with 50 epochs, a batch size of 24, and a learning rate of 0.001 to identify cars and taillights. Figure 2 display the mean average precision (mAP) of the YOLOV5-based on vehicle taillight detection system. The accuracy has certainly increased. Nearly 92.36% of the cases, the system was effective. Figures. 3, system's precision is seen through our work, the model has 91.92% accuracy rate. The system recall, shown in Fig. 4, is nearly 88.96%.

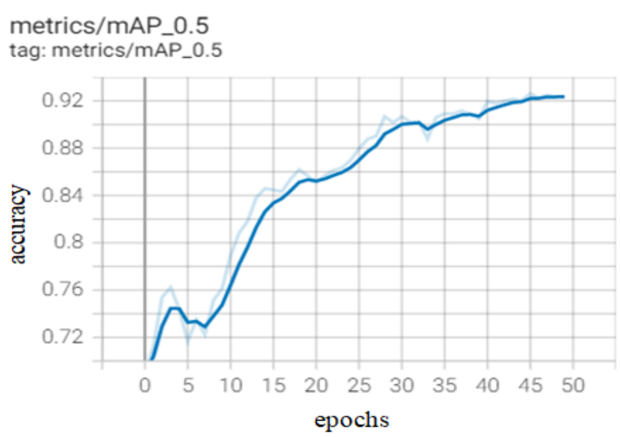

Fig 2. Mean Average Precision

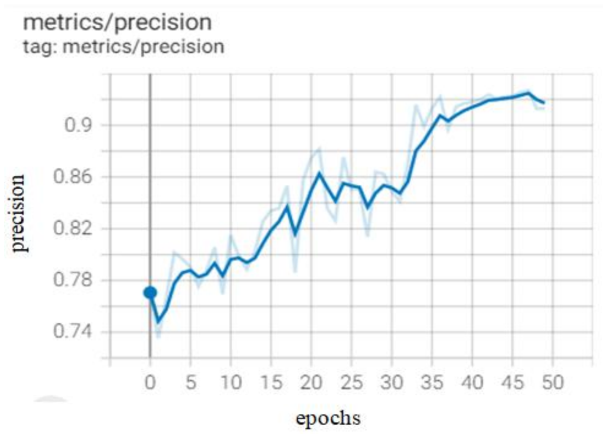

Fig 3. Precision

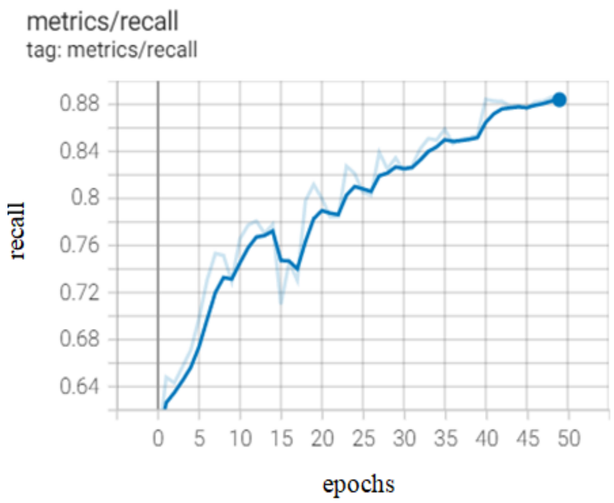

Fig 4. Recall

Fig. 5,6 and 7 represents the training box, class and object losses. Where box loss, class loss and object loss of model decreases with increase in epochs.

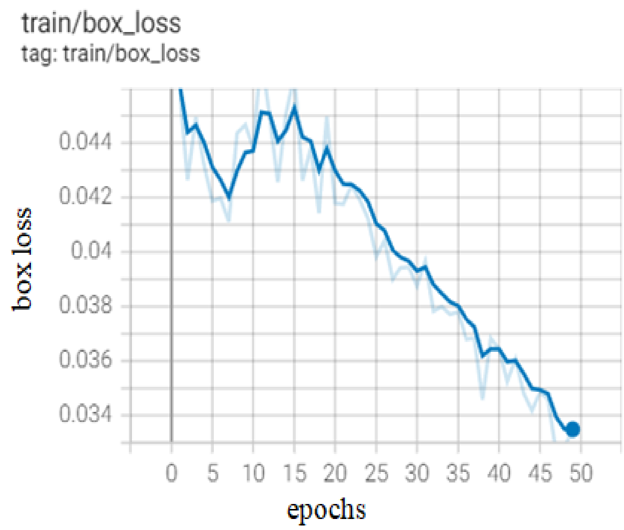

Fig 5. Train box loss

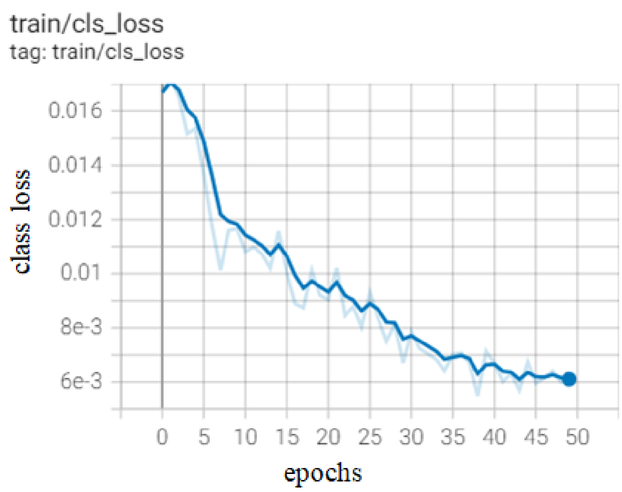

Fig 6. Train class loss

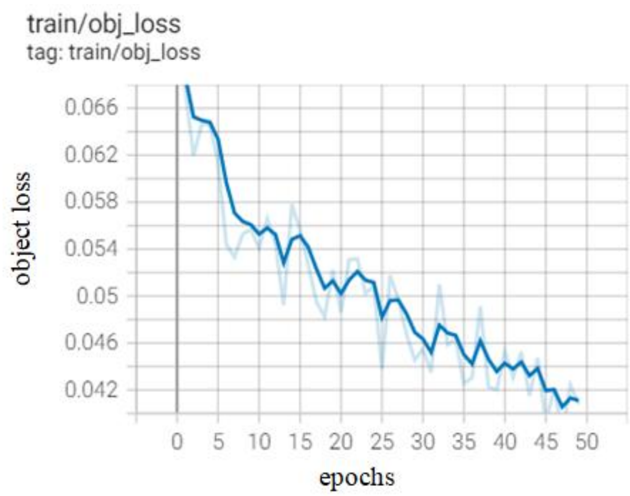

Fig 7. Train object loss

Fig. 8,9 and 10 represents the Validation box, class and object losses. . Where box loss, class loss and object loss of model decreases with increase in epochs. So, the model is fit for detecting vehicle taillight in images and videos.

Fig 8. Validate box loss

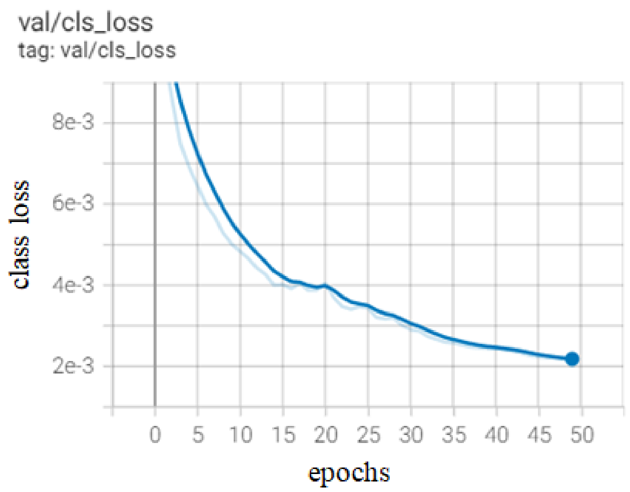

Fig 9. Validate class loss

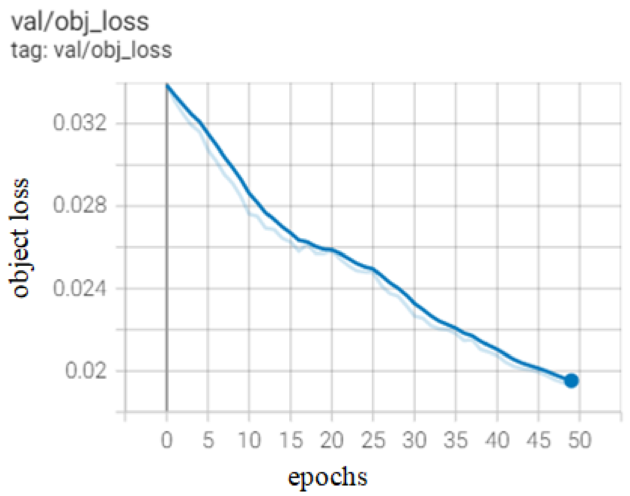

Fig 10. Validate object loss

Fig. 11 represents output obtained when the video input is processed as 98% and 97% confidence ratio. Equation (1), (2) & (3) represents the formulae to calculate the accuracy, precision & recall respectively [12]. Table 1 represents the comparison of evaluation metrics of YOLOV4 and YOLOV5. It clearly concludes that YOLOV5 is the best algorithm that can be used to detect the vehicle taillight.

Fig 11. Output of the proposed system

$$\text{Accuracy} = \frac{TP + FP}{TP + FP + TN + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

TP: True Positive FP: False Positive
TN: True Negative FN: False Negative

TABLE 3. Comparative analysis of current algorithms

Algorithms	**Evaluation Metrics**		
	Accuracy	*Precision*	*Recall*
YOLOV5	92.36%	91.92	88.96
YOLOV3	75.30%	85%	90%

V. CONCLUSION & FUTURE WORK

Real-time detection and identification of taillights can aid in preventing traffic incidents that may result from a driver's disregard for them. In this study we proposed YOLOV5 Algorithms for vehicle taillight detection in autonomous driving. We create a taillight dataset with 4032 images related to car class and taillight class. at the parameters of batch size as 32 and learning rate as 0.001, the model archived 91.92% of mean average precision. Future Work will explore numerous weather circumstances, including cloudy and rainy conditions and also investigate different approaches to calculate the distance between the vehicles.

References

[1] Arunkumar, K. and Vasundra, S. (2022), "Robust multifocus deep neural network for progression prediction on patient trajectory data", International Journal of Intelligent Computing and Cybernetics, Vol. 15 No. 4, pp. 589-598.

[2] J. -G. Wang et al., "Appearance-based Brake-Lights recognition using deep learning and vehicle detection," 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 815-820, doi: 10.1109/IVS.2016.7535481.

[3] F. I. Vancea, A. Daniel Costea and S. Nedevschi, "Vehicle taillight detection and tracking using deep learning and thresholding for candidate generation," 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2017, pp. 267-272, doi: 10.1109/ICCP.2017.8117015.

[4] H.-T. Chen, Y.-C. Wu, and C.-C. Hsu, "Daytime preceding vehicle brake light detection using monocular vision," IEEE Sensors J., vol. 16, no. 1, pp. 120–131, Jan. 2016.

[5] M. Casares, A. Almagambetov and S. Velipasalar, "A Robust Algorithm for the Detection of Vehicle Turn Signals and Brake Lights," 2012 TEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, 2012, pp. 386-391, doi: 10.1109/A VSS.2012.2

[6] G. Zhong et al., "Learning to tell brake lights with convolutional features," in Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC), Nov. 2016, pp. 1558–1563

[7] F. I. Vancea and S. Nedevschi, “Semantic information based vehicle relative orientation and taillight detection,” in Proc. IEEE 14th Int. Conf. Intell. Comput. Commun. Process. (ICCP), Sep. 2018, pp. 259–264.
[8] A. Almagambetov, S. Velipasalar, and M. Casares, “Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras,” IEEE Trans. Ind. Electron., vol. 62, no. 6, pp. 3732–3741, Jun. 201
[9] D. Nava, G. Panzani, and S. M. Savaresi, “A collision warning oriented brake lights detection and classification algorithm based on a mono camera sensor,” in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), Oct. 2019, pp. 319–324.
[10] Q. Li et al., "A Highly Efficient Vehicle Taillight Detection Approach Based on Deep Learning," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4716-4726, July 2021, doi: 10.1109/TITS.2020.3027421.
[11] Yifan Liu ; Bing Hang Lu ; Jingyu Peng ; Zihao Zhang "Research on the Use of YOLOv5 Object Detection Algorithm in Mask Wearing Recognition" World Scientific Research Journal ; 6(11):276-284, 2020.Article in English | Airiti Library | ID: covidwho-994115
[12] Balaram, A., Vasundra, S. (2022). A Review on Machine Learning Techniques to Predict the Reliability in Software Products. In: Gunjan, V.K., Zurada, J.M. (eds) Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications. Lecture Notes in Networks and Systems, vol 237. Springer, Singapore. https://doi.org/10.1007/978-981-16-6407-6_28