Helmet Detection and Reporting System

Pavithra T S[1], Adarsh Y Ulvekar[2]
Department of Computer Applications, PES University
Bangalore, India
[1]pavithrats@pes.edu
[2]adarshulvekar49@gmail.com

Abstract: Helmet is a critical aspect of road safety, as it has been shown to reduce the risk of head injuries. Helmet detection systems can be used to identify rider who are not wearing Helmet. This can help to increase helmet usage and reduce the number of motorcycle injuries. This System will detect the rider with no helmet and record the details of vehicle and will provide the report that will containing details like image, vehicle number, name, phone number, address. This report will be accessed by the traffic controller and also the generated will be sent to the rider.

Keywords: CNN, easyOcr, YOLO v5, OpenCV, PyTorch.

I. INTRODUCTION:

Traffic inspector absence on the scene. One of the primary challenges is the absence of traffic inspectors or law enforcement personnel at crucial locations where violations frequently occur. Without a physical presence, it becomes difficult to effectively monitor and enforce compliance with traffic regulations. Manual monitoring over the vehicle. Human error and fatigue may lead to oversight or inadequate enforcement of violations. Rider unawareness of the rule violation. In many cases, riders may be unaware of their violation of traffic rules, including helmet non-compliance.

The system is designed to analyse input video footage captured by surveillance cameras or monitoring devices to identify instances of motorcycle riders on the road. License plate extraction and recognition functionality is triggered only in cases where helmet non-compliance is detected. Upon detecting a rider without a helmet, the system extracts the vehicle's license plate number from the video footage. System performance considerations. While the system is designed to operate with high accuracy and efficiency, it may not perform perfectly in all scenarios due to various factors such as obscured license plates, challenging lighting conditions, or occluded views. Dependency on vehicle information in the XML. The successful extraction and retrieval of vehicle information, including owner details, relies on the availability and accuracy of data in the system's XML. Addressing data privacy concerns. Given the sensitive nature of the information captured and processed by the system, data privacy considerations are paramount. The system must adhere to strict data protection regulations and protocols to safeguard the privacy and confidentiality of rider information stored within the XML.

To promote motorcycle helmet use and reduce the number of helmet-related accidents and injuries. By automating the detection of helmet violations, the solution aims to increase awareness and adherence to helmet-wearing regulations among motorcycle riders. To automate motorcycle helmet violation detection and streamline enforcement processes. By leveraging advanced technologies such as computer vision and deep learning, the solution automates the detection of helmet non-compliance incidents, eliminating the need for manual monitoring and enforcement efforts. To advance the field of traffic management and create a safer transportation ecosystem through technological solutions. The proposed solution represents a significant advancement in the field of traffic management by harnessing cutting-edge technologies to address road safety challenges. By deploying innovative technological solutions, the solution contributes to the creation of a safer transportation ecosystem, characterized by reduced accident rates, improved compliance with traffic laws, and enhanced overall road safety outcomes.

Two-stage algorithms, exemplified by Faster R-CNN and Mask R-CNN, traditionally boast high detection accuracy. However, they tend to be slower due to their multi-stage architecture, making real-time detection challenging. One-stage algorithms, including popular frameworks like YOLO (You Only Look Once), SSD (Single Shot Detector), RetinaNet, ATSS (Adaptive Training Sample Selection), FCOS (Fully Convolutional One-Stage Object Detection), and RepPoints (based on anchor-free detection), prioritize speed while maintaining a reasonable level of accuracy. Among these, YOLOv5 stands out as the fifth generation of the YOLO algorithm, known for its balance of speed and precision. It offers a streamlined approach to object detection with compact model parameters, making it suitable for real-time applications. The automatic helmet detection system is based on an improved YOLOv5 detector integrated with triplet attention.

II. LITERATURE SURVEY

In Research Paper [1] proposes a system for Helmet, Number Plate Detection, and Stolen Vehicle Recognition using Machine Learning. The model utilizes techniques such as video size/aspect ratio analysis, image classification using CNNs, head region extraction, and feature analysis, as well as optical character recognition (OCR). The system takes CCTV video input, analyses video size and aspect ratio, classifies video content using CNNs, extracts head regions for feature analysis, and employs OCR to extract text like license plate numbers. It aims to automatically detect helmetless motorcycle riders and retrieve license plate information, contributing to road safety and law enforcement efforts.

In Research Paper [2] presents a real-time method for automatic helmet detection of motorcyclists in urban traffic, utilizing the YOLOv5 algorithm introduce a system capable of identifying motorcycles and determining whether riders wear helmets in real-time video surveillance. Key components involve motorcycle detection using YOLOv5, helmet detection using the same algorithm, and outputting real-time data on detected motorcycles and helmet usage. This method contributes to enhancing traffic safety through efficient and automated helmet compliance monitoring.

In Research Paper [3] outlines the creation of an Automatic Detection System for Motorcyclists without Helmets using Machine Learning. This model employs key techniques such as OpenCV for image processing, HAAR cascade for object detection, and Optical Character Recognition (OCR). The system processes input images, identifies helmets, extracts text like license plate numbers, and outputs the detection results. This approach enhances enforcement of safety regulations by automating the detection of helmet and number plate violations.

In Research Paper [4] presents an Automatic Number Plate Detection system for motorcyclists riding without helmets. This model integrates helmet and number plate detection using YOLOv5 and Tesseract OCR, respectively. They preprocess images, detect helmets using object detection algorithms, extract regions of interest, classify detected helmets, and generate alerts using pyWhatkit library. This system enhances safety enforcement by identifying helmet and number plate violations and notifying relevant parties in real-time.

III. PROPOSED SYSTEM

Acceptance of Recorded Video Files and Vehicle Detection. The system accepts recorded video files inputted by admin. The system detects vehicles within the video footage, enabling precise analysis and identification of violations. Helmet Detection Using Trained YOLOv5 Model. Employing state-of-the-art deep learning techniques, specifically the YOLOv5 object detection model, the system accurately identifies instances of helmet non-compliance among motorcycle riders. Extraction of Vehicle License Plate Number Image. Upon detecting a helmet non-compliance incident, the system extracts the vehicle's license plate number image from the video footage. Utilizing advanced image processing algorithms. Conversion of Image to Text Using EasyOCR.

Initially the video file is given as the input to the system which is the passed on to the model which tries to detect the vehicle in the video file if the vehicle is not found then it goes to initial state where the admin again give input else if the vehicle is found then it checks for the helmet detection if helmet is found then it again goes for the initial state else if no helmet is found then it moves further for extracting the number plate.

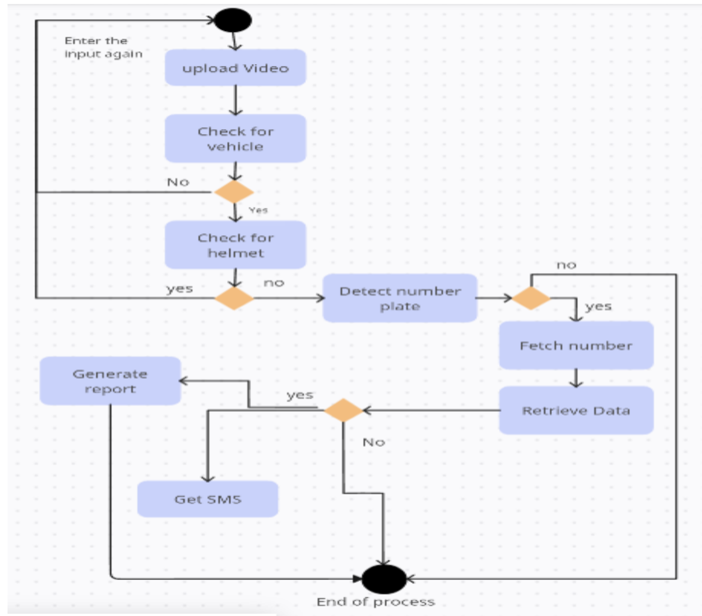


Fig 1. System Design

If the number plate couldn't be extracted, then it goes to initial state. Once the number plate is extracted then it is converted to the text using easyOCR which is then used to fetch the details from the XML. Once the details are fetched it generates the report for the admin

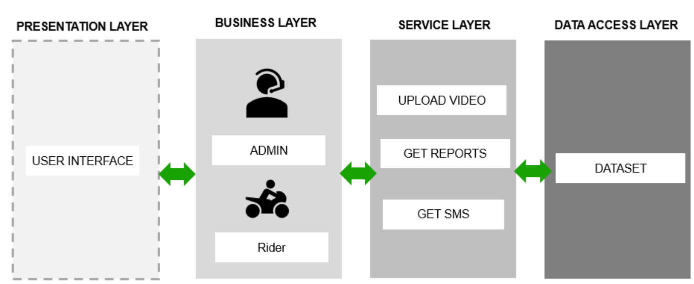


Fig 2. Architecture Diagram

A. Presentation Layer

Serves as the interface between the users and the system, providing a visually appealing and intuitive user experience. It includes components such as the admin dashboard, where administrators can view detected violations, access reports, and manage system settings.

B. Business Layer

Houses the core logic and functionalities of the system, implementing the various algorithms. It is the interface between the user and the services provided by the system through user interface.

C. Services Layer

Acts as the bridge between the system and external services, facilitating communication and integration with services. It also includes components for interfaces.

D. Data Layer

Manages the storage, retrieval, and manipulation of data within the system, ensuring data integrity, security, and efficiency. It utilizes a relational XML to store various types of data, including information about detected violations, rider details, and system configurations.

The extracted license plate number image undergoes Optical Character Recognition (OCR) using EasyOCR technology to convert visual text into machine-readable format. Data Retrieval from XML and Report Generation. Leveraging the extracted license plate number as a unique identifier, the system retrieves pertinent vehicle owner information from a centralized XML. This includes the owner's name, phone number, and other registered details.

A. Data Collection

The dataset collected from the Roboflow website comprises a comprehensive collection of images and corresponding annotations tailored for object detection tasks, likely focusing on detecting helmets. Leveraging Roboflow's platform, the dataset is carefully curated and annotated to ensure accuracy and consistency in bounding box annotations and class labels. This dataset is expected to exhibit diversity, featuring images captured across various environments and conditions relevant to the intended application, thereby enhancing the robustness and generalization capabilities of the trained model. Additionally, Roboflow's preprocessing capabilities enable convenient data augmentation and resizing, facilitating the optimization of dataset diversity and training efficiency.

Fig 3: Bike Dataset

Fig 4: Helmet Dataset

B. Pre-Processing

1) Image Resizing and Normalization: In the img_classify function, resize the input image to a fixed size of 144x144 pixels using the transforms. Resize transformation from the torchvision.transforms module. After resizing, convert the image to a PyTorch tensor and normalize its pixel values to the range [-1, 1] using the transforms.toTensor and transforms.Normalize transformations.

2) Image Permutation and Conversion: Before passing the image to the YOLOv5 model for object detection in the object_detection function, permute the dimensions of the image tensor using permute(2, 0, 1) to match the expected input format of the model (channels-first format).

3) Data Augmentation: While script doesn't explicitly perform data augmentation, but I have applied data augmentation techniques during the training phase to increase the diversity and robustness of the dataset. Common data augmentation techniques include rotation, flipping, and random cropping of images.

Fig 5: Pre-processing Images

C. Training Model

The training data is prepared according to the YOLOv5 format, which typically involves labeling the images with bounding boxes and class labels. The training script assumes that the data is correctly formatted and ready for training.

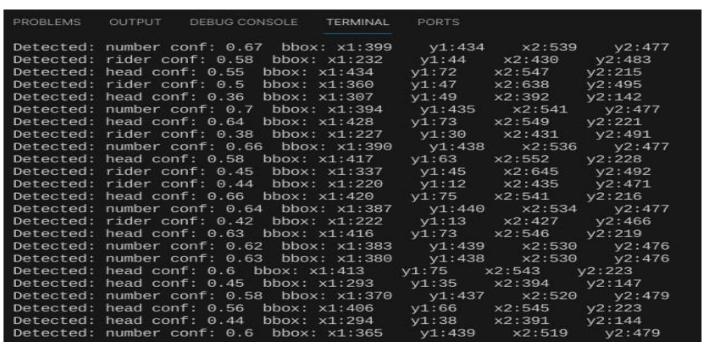
1) *Object Detection:* The script reads frames from a video stream or image file and performs object detection using the trained YOLOv5 model. Detected objects are classified into three categories: riders, heads, and number plates.

2) *Helmet Classification:* For each detected head, the script extracts the region of interest (ROI) and performs helmet classification using a separate classifier model. Based on the classification result, the script determines whether a helmet is present or absent.

3) *Saving Results:* The script saves the processed video with bounding boxes drawn around detected objects and helmet classification results. Additionally, it saves images of riders and number plates if detected.

4) *Displaying Results:* The script display the processed video frame by frame, showing the detection and classification results in real-time.

D. Performance Evaluation



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Detected: number conf: 0.67  bbox: x1:399    y1:434    x2:539    y2:477
Detected: rider conf: 0.58  bbox: x1:232    y1:44    x2:430    y2:483
Detected: head conf: 0.55  bbox: x1:434    y1:72    x2:547    y2:215
Detected: rider conf: 0.5  bbox: x1:360    y1:47    x2:638    y2:495
Detected: head conf: 0.36  bbox: x1:307    y1:49    x2:392    y2:142
Detected: number conf: 0.7  bbox: x1:394    y1:435    x2:541    y2:477
Detected: head conf: 0.64  bbox: x1:428    y1:73    x2:549    y2:221
Detected: rider conf: 0.38  bbox: x1:227    y1:30    x2:431    y2:491
Detected: number conf: 0.66  bbox: x1:390    y1:438    x2:536    y2:477
Detected: head conf: 0.58  bbox: x1:417    y1:63    x2:552    y2:228
Detected: rider conf: 0.45  bbox: x1:337    y1:45    x2:645    y2:492
Detected: rider conf: 0.44  bbox: x1:220    y1:12    x2:435    y2:471
Detected: head conf: 0.66  bbox: x1:420    y1:75    x2:541    y2:216
Detected: number conf: 0.64  bbox: x1:387    y1:440    x2:534    y2:477
Detected: rider conf: 0.42  bbox: x1:222    y1:13    x2:427    y2:466
Detected: head conf: 0.63  bbox: x1:416    y1:73    x2:546    y2:219
Detected: number conf: 0.62  bbox: x1:383    y1:439    x2:530    y2:476
Detected: number conf: 0.63  bbox: x1:380    y1:438    x2:530    y2:476
Detected: head conf: 0.6  bbox: x1:413    y1:75    x2:543    y2:223
Detected: head conf: 0.45  bbox: x1:293    y1:35    x2:394    y2:147
Detected: number conf: 0.58  bbox: x1:370    y1:437    x2:520    y2:479
Detected: head conf: 0.56  bbox: x1:406    y1:66    x2:545    y2:223
Detected: head conf: 0.44  bbox: x1:294    y1:38    x2:391    y2:144
Detected: number conf: 0.6  bbox: x1:365    y1:439    x2:519    y2:479
```

Fig 6: ROI Detection Conf

The designed model make use of ROI method which in return provides the coordinate of the detected images. Detected: rider conf: 0.58 bbox: x1:232 y1:44 x2:547 y2:215 This line suggests that the program detected a rider with a confidence score of 0.58 (possibly out of 1.0). The bounding box coordinates are also provided. Detected: head conf: 0.64 bbox: x1:428 y1:73 x2:549 y2:221 y2:477 This line indicates that the program detected a head with a confidence score of 0.64.

IV. METHODOLOGY

A. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily designed to process and analyse visual data, such as images and videos. CNNs are characterized by their ability to automatically and adaptively learn spatial hierarchies of features from raw input data.

For a 2D image H and 2D Filter(kernel) F

1.) Convolution Operation: G=H * F,
$G[i, j]=\sum_{u=-k}^{k} \sum_{u=-k}^{k} H[u, v]F[i-u, j-v]$

2.) Correlation Operation: G=H Ø F,
$G[i, j]=\sum_{u=-k}^{k} \sum_{u=-k}^{k} H[u, v]F[i+u, j+v]$

B. Open-Source Computer Vision Library (OpenCV)

It is an open-source computer vision and machine learning software library designed for real-time applications. It provides a wide range of functionalities for image and video processing.

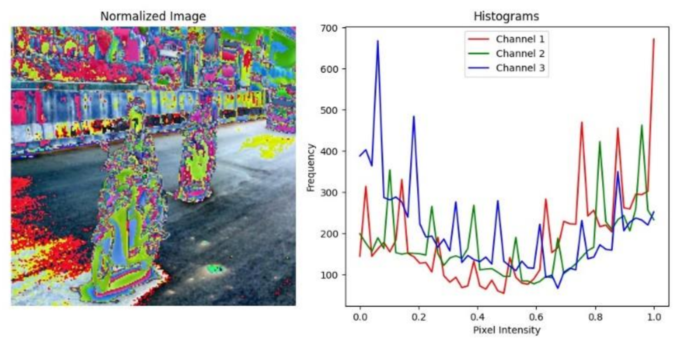


Fig 7: Sample Image

C. You Only Look Once (YOLO v5)

It is an open-source object detection framework that builds on the success of previous versions (YOLOv1, YOLOv2, YOLOv3, and YOLOv4). It was developed by Ultralytics and released in mid-2020. YOLO stands for "You Only Look Once," referring to its ability to perform object detection in real-time by processing the entire image in a single forward pass through a convolutional neural network (CNN).

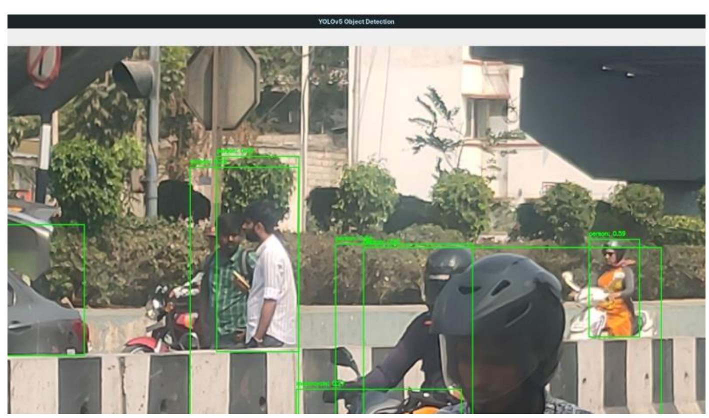


Fig 8. Bounding Box

D. PyTorch

It is a popular deep learning framework built for Python. While PyTorch doesn't rely on a single overarching formula, it leverages several key mathematical concepts for deep learning tasks. PyTorch is used to perform forward passes on pre-processed head regions from the video frame using the helmet classification model. It retrieves the predicted class (helmet or no helmet) and the corresponding confidence score for each head region.

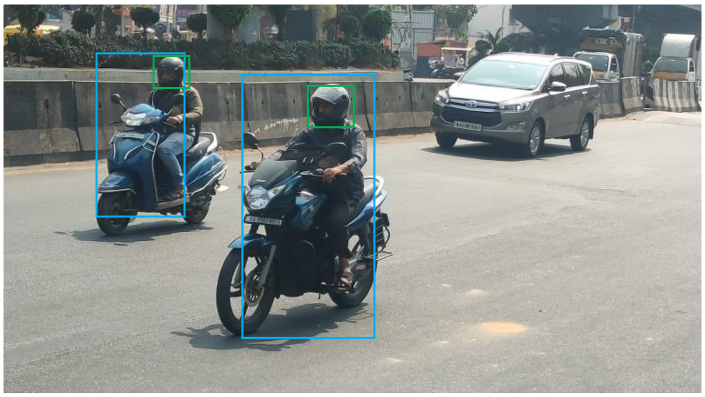


Fig 9: Head Region Extraction

E. Imutils ,NumPy, PyPlot

Image processing in Python often involves a powerful trio. Imutils acts as a handy layer on top of OpenCV, simplifying common image manipulation tasks. NumPy, the workhorse of scientific computing, efficiently manages the numerical data that underlies images. Finally, pyplot, a part of Matplotlib, takes the processed image data from NumPy and transforms it into visualizations like charts and plots.

Fig 10: Gray-scale Plotting.

F. Easy Optical Recognition (easyOCR)

It is a Python library designed to perform Optical Character Recognition (OCR) tasks with ease. It utilizes pre-trained deep learning models to extract text from images and documents, making it suitable for a wide range of applications such as document scanning, text extraction from images.

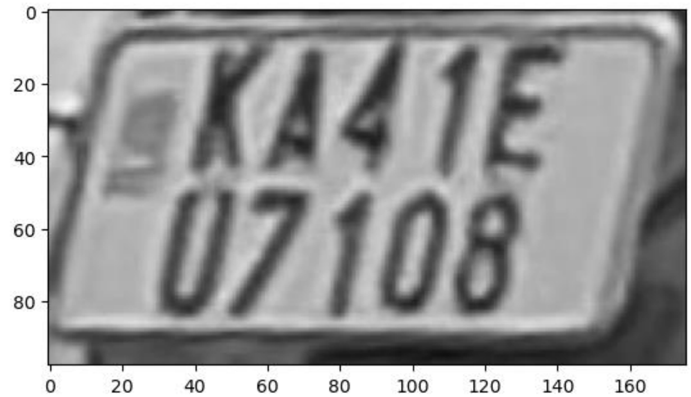


Fig. 11 Edge Extraction

```
easyOcr Accuracy:
Text: KA41E U7108 , probablity:0.7510244193586483
```

Fig . 12 :Text Extraction

The probability of the Fig,2 after processing of easyOCR is given above.

G. Twilio API

Twilio is a cloud communications platform that provides APIs for developers to programmatically integrate voice, SMS, and other communication services into their applications. It offers a wide range of features and functionalities for building scalable and reliable communication solutions.

Implementing a system to send SMS notifications to riders detected without helmets using helmet detection technology involves several key steps. Developed a robust helmet detection system capable of analysing video feeds to identify riders not wearing helmets accurately. For violation is detected, the system triggers an event to send an SMS notification to the rider's phone number. Integration with Twilio's API is essential for sending SMS messages programmatically. This integration requires obtaining Twilio credentials, including the account SID, authentication token, and a Twilio phone number. With these credentials, application can authenticate and interact with Twilio's services to send SMS messages.

V. CONCLUSION:

A reliable method to identify and flag motorcycle riders who are not wearing helmets is provided by the model described in this research. The ingenious approach to train the neural network using Transfer Learning technique along with the usage of open-source libraries in the code allows for the development of a cost-effective system to assist personnel to catch violators. YOLO algorithm used in our model shows us promising results, demonstrating its applicability in real world situations. The outcome of this study can be utilized as the foundation for integrating this technology with fully autonomous systems to enable their operation in a wholly independent manner.

ACKNOWLEDGE

I extend my heartfelt gratitude to my Guide Ms. Pavithra T S , for their unwavering guidance, encouragement, and invaluable insights throughout the duration of this research endeavor. Their expertise and mentorship have been instrumental in shaping the direction and quality of our work. Furthermore, I am deeply thankful to PES University for providing access to the necessary resources, facilities, and infrastructure essential for conducting our experiments effectively.

REFERENCE

[1] Gauri Marathe1, Pradnya Gurav2, Rushikesh Narwade3, Vallabh Ghodke4, Prof. S. M. Patil5(May 2022)” Helmet Detection and Number Plate Recognition using Machine Learning” May 2022 | IJIRT | Volume 8 Issue 12 | ISSN: 2349-6002.

[2] Lokesh Allamki1, Manjunath Panchakshari2, Ashish Sateesha3, K S Pratheek4(Dec 2019)” Helmet Detection using Machine Learning and Automatic License Plate Recognition” Volume: 06 Issue: 12 | IRJET| Dec 2019 e-ISSN: 2395-0056 p-ISSN: 2395-0072

[3] A. Adam, E. Rivlin, I. Shimshon, and D. Reinitz, “Robust realtime unusual event detection using multiple fixed location monitors,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pp. 555–560.

Volume: 06 Issue: 12| Dec2019 e-ISSN: 2395-0056 p-ISSN: 23950072

[4] Romuere R.V.e Silva, Kelson R.T. Aires, Rodrigo de M. S. Veras Year: 2021 “Helmet Detection and Number Plate Recognition for Safety and Surveillance System”/IRJMETS/ Volume:03/Issue:02/February 2021 e-ISSN: 2582-5208

[5]. Fahad A Khan, Nitin Nagori, Dr. Ameya Naik Year: 2020 “Helmet and Number Plate detection of Motorcyclists using Deep Learning and Advanced Machine Vision Techniques”/ 01 September 2020 /IEEE e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 24, Issue 2

[7] A S Mohammed Shariff, Raghav Bhatia, Raghwendra Kumar, Sarthak Jha, “Vehicle Number Plate Detection using Python and Open CV”,2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) DOI: 10.1109/ICACITE51222.2021

[8] Mahesh Babu K, M V Raghunadh, “Vehicle Number Plate Detection and Recognition using Bounding Box Method”, International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) 2016.

[9] Amitava Choudhary, Alok Negi, “A new zone based algorithm for detection of license plate from Indian Vehicle”, Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)2016.

[10] Prof. M.V. Sadaphule, Kshitij Patil, Aniruddha Patil, Kunal Waghmare, Supriya Nikale “Automatic Number Plate Recognition System Using CNN”, JETIR (Journal of Emerging Technologies and Innovative Research).