Automatic Traffic Rules Violation Detection and Number Plate Recognition

Arun P
dept of Electronics and Communication Engineering St.Xavier's Catholic College of Engineering
Kanyakumari, Tamil Nadu
arunmass1603@gmail.com

Aakash S
dept of Electronics and Communication Engineering St.Xavier's Catholic College of Engineering
Kanyakumari, Tamil Nadu
aakashaakazh@gmail.com

Jebin Mon M P
dept of Electronics and Communication Engineering St.Xavier's Catholic College of Engineering
Kanyakumari, Tamil Nadu
jebinmon1818@gmail.com

Abilesh V
dept of Electronics and Communication Engineering
St.Xavier's Catholic College of Engineering
Kanyakumari, Tamil Nadu
jeevithajeba618@gmail.com

Mary Little Flower B.E,M.E
dept of Electronics and Communication Engineering St.Xavier's Catholic College of Engineering
Kanyakumari, Tamil Nadu
mary@sxcce.edu.in.com

***Abstract*—As road traffic continues to grow and traffic rule violations become more common, there's a growing need for smart systems that can automatically detect and report such incidents. This project introduces an automated solution for detecting traffic violations and recognizing license plates using deep learning, convolutional neural networks (CNNs), and optical character recognition (OCR). The system is capable of identifying common offenses such as riding without a helmet, triple riding, and running a red light, by analyzing real-time video footage. Using object detection models like YOLO, it accurately spots violators and isolates the license plate area from the video frames. Tools like EasyOCR are then used to read and extract the alphanumeric details from the plates. Once a violation is detected, the system automatically sends an email—with the offense details and a snapshot of the incident—using Python's smtplib library. This solution is designed to support traffic enforcement agencies by making violation tracking more efficient, reliable, and automated..**

***Keywords*—Deep Learning,CNN and OCR..**

I. Introduction

With the rapid growth in the number of vehicles on the road, traffic rule violations and accidents have also increased. Monitoring and enforcing these rules manually is not only time-consuming but also prone to errors—especially in busy cities. Traditional traffic monitoring systems often struggle to catch and report violations in real-time, which makes it harder to maintain road safety.

To solve this problem, this project introduces an automated system that can detect common traffic violations such as riding without a helmet, triple riding, and signal jumping by analyzing video footage in real-time. Using deep learning and convolutional neural networks (CNNs), the system can accurately identify violators. It then uses object detection models like YOLO to find the vehicle's license plate, and applies OCR (Optical Character Recognition) with tools like EasyOCR to read the plate number

Once a violation is detected, the system captures an image of the incident and sends an email with all the details like photo of the violator and evidence to the appropriate authorities using Python's smtplib and store them in a database,where the traffic police can able to access. This makes the entire process—from detection to reporting—fully automatic.

The main goal of this project is to help traffic police and enforcement agencies by providing a smart and efficient tool to monitor violations, improve road safety, and reduce manual effort.

II. Motivation

Traffic violations are a major reason behind many road accidents and deaths around the world. Even though traffic rules exist, people often break them—like riding without helmets, carrying too many passengers, or jumping red lights. It's hard for traffic police to catch every violation, especially with so many vehicles on the road and limited staff. Manual monitoring is also tiring and can lead to mistakes.

This project is motivated by the need to solve these problems through automation. Using deep learning and computer vision, we can build a system that works all the time, detects violations accurately, and doesn't get tired like humans do. It can instantly spot a violation, capture the vehicle's number plate using OCR, and save the image as proof.

The system also sends all this information directly to the authorities through email, making the process faster and more efficient. It supports the idea of smart cities by bringing modern technology into traffic management.

Overall, the goal is to create a smart, affordable, and scalable system that helps traffic police, improves safety, and saves lives on the road.solution, suitable for challenging image conditions.

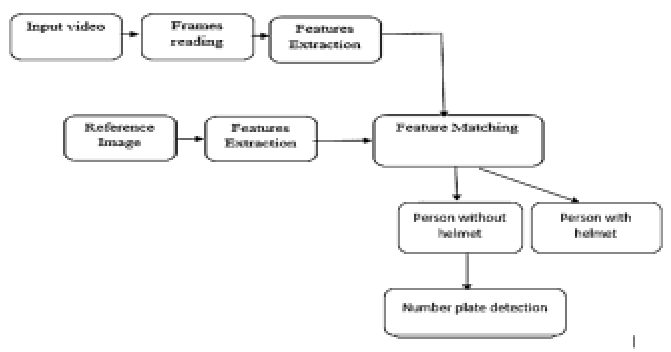
III. Proposed System

This project presents an intelligent system that automates the detection of traffic violations and the recognition of vehicle number plates using advanced technologies like deep learning, computer vision, and Optical Character Recognition (OCR). The system processes live video feeds from CCTV cameras or recorded traffic footage in real-time. It identifies common traffic violations, such as not wearing a helmet, triple riding on two-wheelers, and jumping red lights, by using a deep learning model like YOLO (You Only Look Once) that can detect multiple objects accurately and quickly in a single frame.

Once a violation is detected, the system automatically locates the vehicle's license plate and uses OCR tools like EasyOCR to extract the alphanumeric characters from the plate. The system then captures an image of the violation, along with the vehicle's number plate, as evidence. To streamline the process, it automatically sends an email notification to the relevant traffic authorities, which includes the violation details like photo, license plate number, violation type, and the time and date of the incident.

This system is scalable, meaning it can be deployed in various locations, and can be integrated into broader smart city infrastructure for more efficient traffic management. Overall, the system helps reduce the burden on traffic police, speeds up violation reporting and enforcement, and contributes to improving road safety and creating smarter cities.

IV. Block Diagram



This block diagram explains the working process of the intelligent traffic violation detection system in a simple and clear way:

Input Video: The system starts by taking a live video feed or a recorded video from a traffic camera.

Frame Reading: The video is broken down into individual frames so they can be analyzed one by one.

Feature Extraction (from Video Frames): Important visual features like people, vehicles, helmets, etc., are detected from each frame using image processing and deep learning techniques.

Reference Image and Feature Extraction: A reference image (for example, of a person wearing a helmet) is used for comparison. Features are also extracted from this image.

Feature Matching: The features from the video frames are compared with the reference features to identify violations. For example, it checks whether the person on a bike is wearing a helmet or not.

Person Classification: If the person is wearing a helmet, no violation is recorded.If the person is not wearing a helmet, it is marked as a violation.

Number Plate Detection: For those violating traffic rules, the system automatically detects the vehicle's number plate for identification.

This flow allows the system to continuously monitor traffic, detect specific violations like riding without a helmet, and capture necessary information like license plate numbers for reporting.

A. *Feature Extraction*

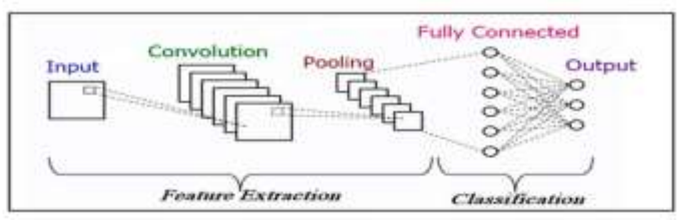


Fig.1. Feature Extraction

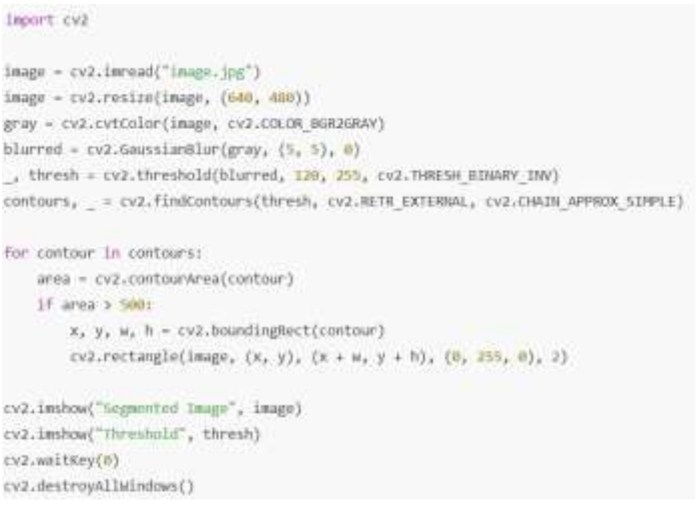
In feature extraction, the computer tries to understand the important parts of an image. First, it uses a process called convolution, where it slides a small window over the image to detect simple patterns like edges, shapes, and textures—similar to how our eyes notice outlines or corners. Then, it uses pooling, which reduces the size of the image but keeps the most important information. This is like zooming out while still being able to see what matters. Together, these steps help the computer focus on key features in the image, such as whether a person is wearing a helmet, identifying a face, or spotting a number plate, without getting confused by unnecessary details.

Gray Scale Conversion:

Grayscale conversion turns a color image into shades of black and white by removing color. In this project, it helps the system focus on shapes and patterns instead of colors, making the processing faster and easier for tasks like detecting helmets, faces, or number plates.

Grayscale = (0.299 * Red) + (0.587 * Green) + (0.114 * Blue

Segmentation Process:



```
import cv2

image = cv2.imread("image.jpg")
image = cv2.resize(image, (640, 480))
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
_, thresh = cv2.threshold(blurred, 120, 255, cv2.THRESH_BINARY_INV)
contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

for contour in contours:
    area = cv2.contourArea(contour)
    if area > 500:
        x, y, w, h = cv2.boundingRect(contour)
        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.imshow("Segmented Image", image)
cv2.imshow("Threshold", thresh)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Fig.2. Segmentation Code

Segmentation is like breaking down an image into smaller, meaningful parts so that a computer can better understand what it's looking at. In this project, segmentation helps the system focus on key areas like a person's head (to check for a helmet) or a vehicle's number plate. The process starts with a raw image or video frame, which goes through preprocessing — converting it to grayscale and applying Gaussian blur to reduce noise and make features clearer. Then, techniques like thresholding or edge detection are used to highlight the outlines of important objects. After that, the system selects only the relevant areas, like the region where a number plate or face is likely to appear. These regions are isolated using contour detection or masking, which helps to separate the object from the background. This makes it easier for the system to analyze just the necessary parts of the image, improving the accuracy of helmet detection or number plate recognition. In short, segmentation lets the system "zoom in" on what really matters and ignore the rest..

Pooling Layers:

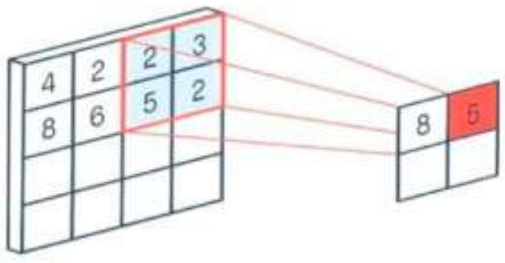


Fig. 3. Max pooling layer.

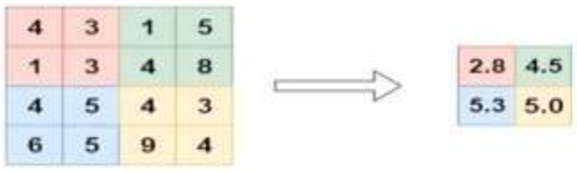


Fig. 4. Average pooling layer.

Pooling layers are a key part of convolutional neural networks (CNNs) that help simplify the information in an image while keeping the important details. Instead of analyzing every tiny pixel, pooling takes a small region of the image and summarizes it—usually by picking the most prominent value (max pooling) or averaging the values (average pooling). This process reduces the overall size of the data, making the system faster and less likely to get confused by small changes in the image. Think of it like zooming out a little—you're seeing less detail, but you still get the main idea. Pooling helps the model stay focused on the big picture, like identifying if someone is wearing a helmet or spotting a number plate, without being distracted by background noise or small variations.

Proposed Algorithm:

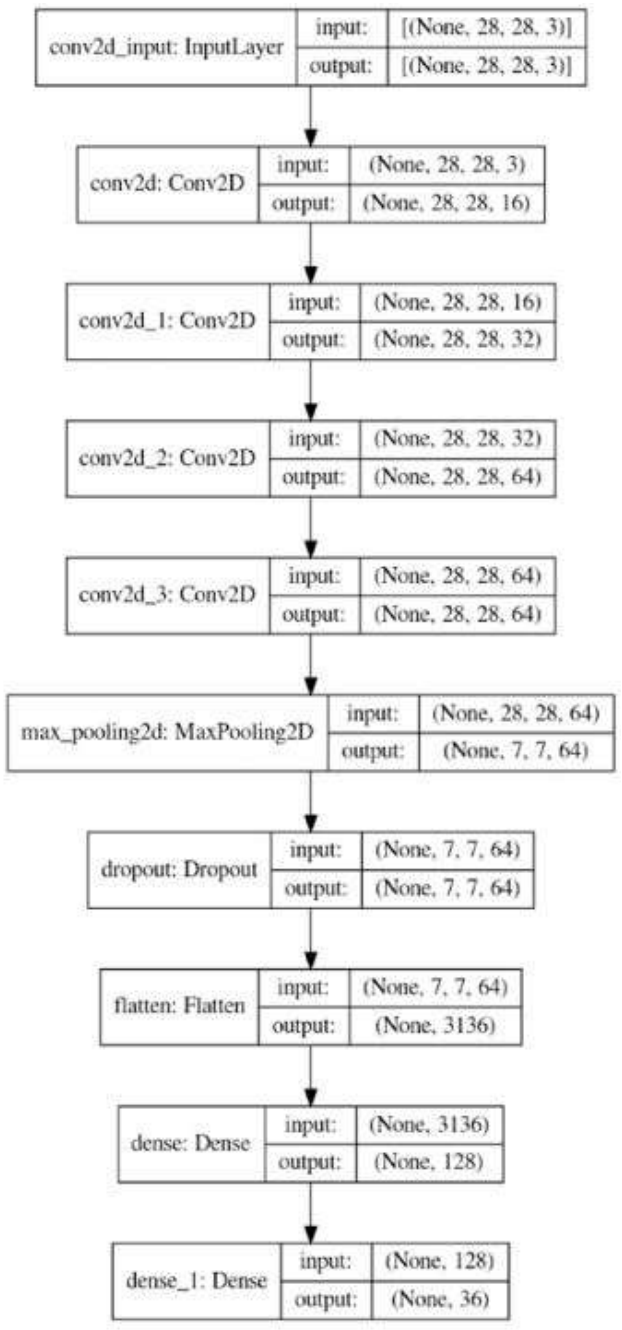


Fig.5.CNN Model

This flowchart shows how a Convolutional Neural Network (CNN) processes an image step by step. It starts with an input image of size 28×28 with 3 color channels (RGB). The image passes through four convolutional layers, each time increasing the number of filters to help the model learn more complex features like edges, shapes, or textures. After that, a max pooling layer reduces the size of the data, making it smaller but still meaningful. A dropout layer is added to prevent overfitting, followed by a flatten layer that converts the data into a one-dimensional format. Finally, it goes through two dense (fully connected) layers, which help in making the final prediction. In the end, the model outputs 36 classes, possibly representing categories like letters, digits, or objects..

Image Thresholding:

Thresholding is a simple way to separate objects in an image from the background. It works by choosing a certain pixel value called a "threshold." If a pixel is brighter than this value, it turns white; if it's darker, it turns black. This creates a clear black-and-white version of the image, making it easier to find shapes or objects. It's especially useful when we want to detect things like text, edges, or patterns in an image. Thresholding is often used as the first step before deeper image processing tasks because it simplifies the image and highlights the important parts. it smaller

Character Segmentation:

Character segmentation is a vital step in optical character recognition (OCR), where the objective is to convert images or scanned text into machine-readable format. This process involves isolating individual characters by identifying the spaces and boundaries between them in the text. The system analyzes the input image, detecting gaps, spaces, and sometimes even lines, to separate the characters accurately. Once segmented, the characters can be recognized and matched to corresponding letters, numbers, or symbols. Proper segmentation is crucial, as errors such as merging or splitting characters incorrectly can lead to recognition mistakes, making the final output unreliable. Therefore, accurate character segmentation plays a key role in ensuring the success of OCR systems.

Character Recognition:

Character recognition using Convolutional Neural Networks (CNNs) is a widely used approach for identifying letters or digits from images, especially in tasks like handwriting recognition or document scanning. CNNs are well-suited for this because they can automatically learn visual features from the input images, such as edges, curves, and shapes that define each character. The process involves passing the image through several layers in the network—starting with convolutional layers that detect features, followed by pooling layers that simplify the data, and finally fully connected layers that classify the character. At the end, the network predicts which character it is by comparing the features it has learned with known patterns. CNNs are highly accurate and efficient, making them a popular choice in OCR systems and other text-related image processing applications.

CNN Model:

A CNN (Convolutional Neural Network) model is a type of deep learning architecture specifically designed for processing and analyzing visual data, such as images. It mimics the way humans recognize visual patterns by automatically learning features from raw pixel data.

Input Layer:-Receives the raw image data (e.g., a 28x28 pixel grayscale image of a handwritten character).

Convolutional Layers:-These layers apply filters (or kernels) that scan across the image and extract important features like edges, curves, and textures. Each filter captures a different pattern.

Activation Function (ReLU):-After convolution, a ReLU (Rectified Linear Unit) function is usually applied to introduce non-linearity, helping the model learn complex patterns.

Pooling Layers:-These layers reduce the size of the feature maps (downsampling), which helps to lower the computational load and focus on the most important information.

Fully Connected Layers:-After several convolution and pooling layers, the output is flattened into a vector and passed through one or more fully connected (dense) layers, similar to traditional neural networks.

Output Layer:-Typically ends with a softmax activation function that gives probabilities for each class (e.g., A-Z or 0-9 in character recognition).

YOLOv5:

YOLOv5 can be a powerful tool in a character recognition project, especially when the characters need to be located in complex or cluttered images, like license plates, street signs, or handwritten forms. Instead of manually segmenting characters, YOLOv5 can automatically detect and draw bounding boxes around each individual character or word in an image. This makes it ideal for the detection phase of the pipeline. Once the characters are detected and cropped out, they can then be passed to a CNN model trained to recognize and classify each one. This combination—using YOLOv5 for detection and CNN for recognition—creates an efficient and accurate character recognition system. YOLOv5 is particularly useful because it's fast, works well in real-time applications, and performs accurately even when dealing with small or overlapping characters.

Epochs:

In deep learning, an epoch refers to one complete cycle through the entire training dataset. In simple terms, it's like teaching the system by showing it all the training data once. The more epochs we run, the more the model learns and improves its accuracy. However, too many epochs can lead to overfitting, where the model performs well on training data but not on new, unseen data. So, finding the right number of epochs is important for good performance.

Sl.No	Parameters	CNN
0	Accuracy	0.9850
1	Precision	0.9803
2	Recall	0.9780
3	F1 Score	0.9791
4	Loss	0.0425

Table:1 Performance Measurement Table

License Plate extraction and Pre-processing:

Pre-processing plays a crucial role in license plate recognition by preparing the image for accurate character detection and recognition. It starts by converting the image to grayscale, which simplifies the data by removing color information while keeping the important details. Next, techniques like blurring are used to reduce noise, and edge detection helps to highlight the outlines of the license plate and its characters. Sometimes, thresholding is applied to make the text stand out more clearly against the background. Morphological operations like dilation or erosion can also be used to enhance the shapes of the characters or remove small unwanted elements. Once the license plate area is clear and clean, it can be cropped and resized for further steps like character segmentation and recognition. Overall, pre-processing makes the image cleaner and more consistent, helping models like YOLOv5 and CNNs work more accurately and efficiently.

Dataset:

In a license plate recognition project, the dataset plays a key role in training the models for both detection and recognition. Usually, two types of datasets are used. The first is for license plate detection and includes images of vehicles with labeled bounding boxes around the plates. This helps models like YOLOv5 learn to accurately find the location of license plates in different conditions, such as various angles, lighting, or weather. Datasets like OpenALPR, AOLP, and CCPD are commonly used for this purpose. The second type of dataset is for character recognition and contains individual characters (A–Z, 0–9) cropped from license plates. These are used to train CNN models to identify each character correctly. Datasets like Chars74K or custom sets created by manually labeling

Precision:

It a parameter can be calculated as the ratio between the true positives and all the positives including false positive and true positive. In our proposal it would be the measure of the license plates that has been correctly identified rates character out of all the license plates Mathematically it can be represented as shown in Equation

Precision = True Positive(TP) / (True Positive (TP)+ False Negative(FP)

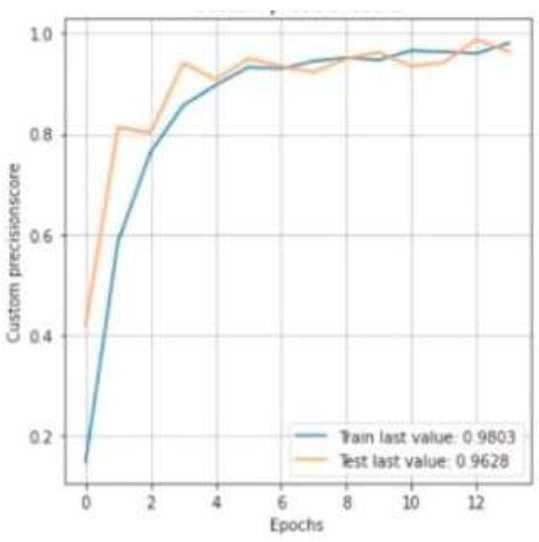


Fig.6. Epochs vs Accuracy

Recall:

It is the parameter which measures that the proposed model is correctly identifying the True Positives. In our proposal it tells us that how many correct license-plates have been recognized. Mathe-matically it can be represented as shown in equation

Recall = True Positive(TP) / (True Positive (TP)+ False Negative(FN)

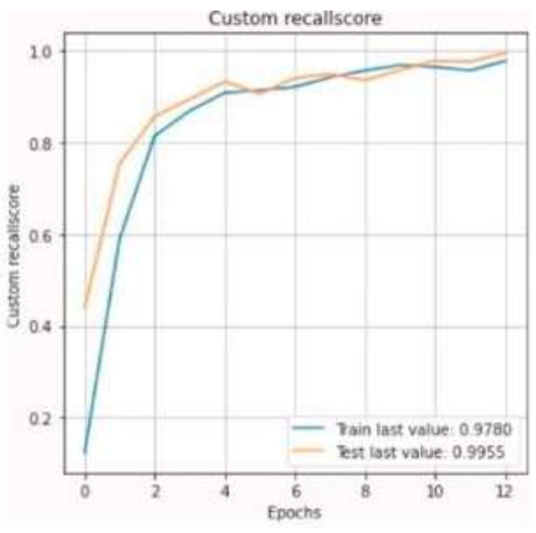


Fig.7. Epochs vs Recall

Accuracy:

It is the parameter which can be calculated as the ratio of total number of correct predictions (True Positives + True Negatives) to the total number of productions (True Positives + True Negatives + False Positives + False Negatives). In our proposal, accuracy tells us how accurately and precisely the license plate has been detected and recognized by the proposed models. Mathematically it can be represented as shown in equation

Accuracy = True Positive (TP) + True Negative(TN) / (True Positive(TP) + False Positive (FP)+True Negative (TN) - False Negative(FN)

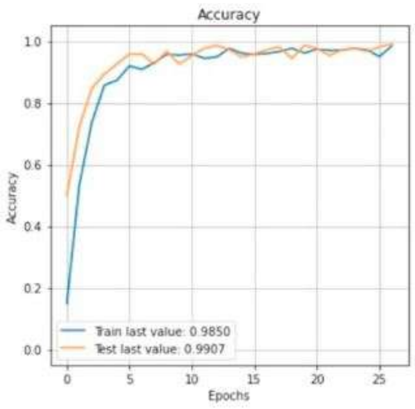


Fig.7. Epochs vs Accuracy

F1 Score:

This parameter gives the result in the basis of Precision and Recall. In some applications, Precision is an important parameter and, in some applications, Recall has an important role for decision making. F1 score is calculated as the harmonic mean of the Precision and Recall values, emphasizing the importance of both the parameters. Mathematically it can be represented as

F1 Score = 2 * Precision * Recall / Precision + Recall

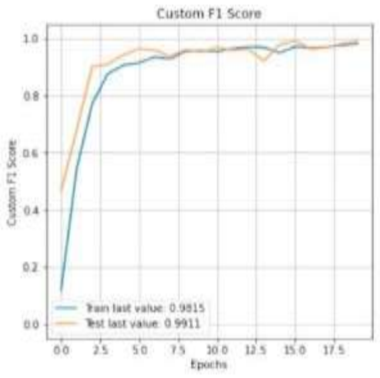


Fig.8. Epochs vs F1 Score

Loss:

Loss or Log-Loss is a parameter which is able to indicate that how close the prediction probability is to the corresponding true value or actual value. If the prediction probability diverges from the true value or actual value then the loss will be more .

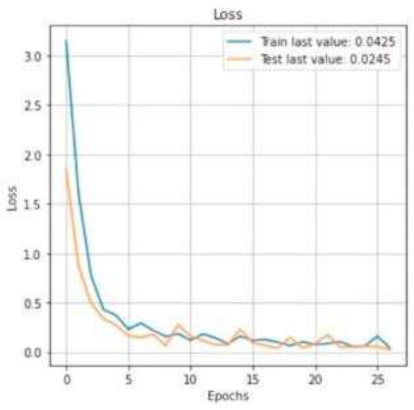


Fig.9. Epochs vs Loss

Result:

The result of this license plate recognition project is a smooth and reliable system that can automatically detect and read license plates from images or videos. Using YOLOv5, the system first identifies and highlights the license plate area on a vehicle with high accuracy, even in challenging conditions like poor lighting, different angles, or busy backgrounds. Once the plate is detected, it's cropped and passed to a CNN model, which then recognizes and reads each character on the plate. The final output is the full license plate number in text format, such as "MH12AB1234". Overall, the system works efficiently and delivers accurate results in real time, making it useful for practical applications like traffic surveillance, automatic toll collection, and smart parking systems.

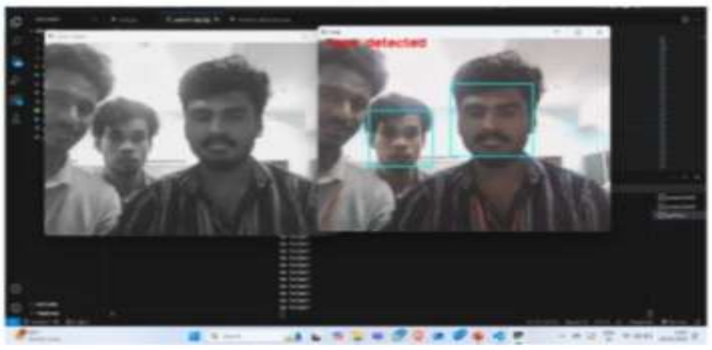


Fig.8. Output of Real-Time Video Capture

The Code executed successfully and video was captured in Real time.

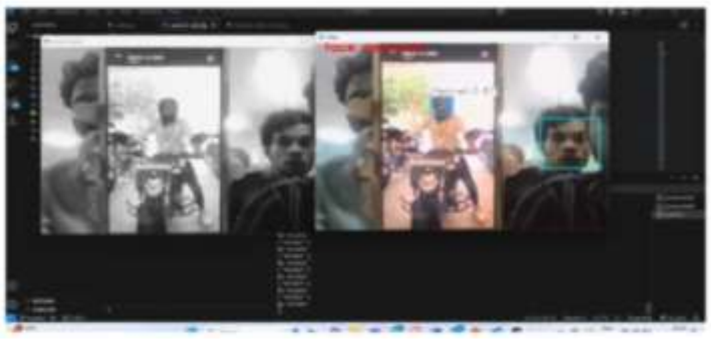


Fig.9. Output of Real-Time Video Capture with Helmet

After that ,the system identifies whether the person is wearing Helmet or Not

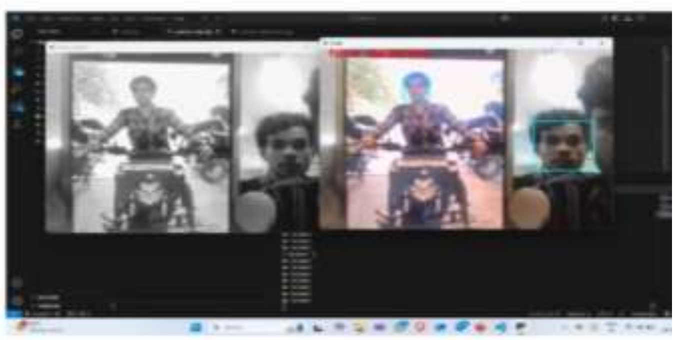


Fig.10. Output of Real-Time Video Capture without Helmet

If The Person not wearing helmet,the number plate will be recognized and Mail will be sent to the violator mail through Smtplib.

V. Conclusion

In conclusion, this license plate recognition project successfully combines the strengths of YOLOv5 and CNN to create an accurate and efficient system for detecting and reading vehicle license plates. By using YOLOv5, the system can quickly and reliably locate license plates in various environments, while the CNN model handles the character recognition with high precision. Through proper pre-processing and the use of suitable datasets, the system is able to perform well in real-world scenarios. This project demonstrates how deep learning techniques can be effectively applied to automate tasks like vehicle monitoring, toll collection, and parking management, offering a smart and scalable solution for modern transportation needs.

References

[1] X. Ascar Davix, C. Seldev Christopher, D. Judson, Detection of the vehicle License Plate using a kernel density with default search radius algorithm filter, in: International Journal of Light and Electron Optics, Optik, Elsevier, 2018, pp. 1-8, 2020.

[2] P. Ravi Kiran Varma, Srikanth Ganta, B. Hari Krishna, S.V.S.R.K. Praveen, A novel method for Indian vehicle registration Number Plate Detection and recognition using image processing techniques detection and recognition using image processing techniques, in: International Conference on Computational Intelligence and Data Science (ICCIDS 2019) vol. 167, Elsevier, Procedia Computer Science, 2020, pp. 2623-2633.

[3] Hanit Karwal, Akshay Girdhar, Vehicle Number Plate Detection System for Indian Vehicles", IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 8-12,

[4] P. William, A. Shrivastava, N. Shunmuga Karpagam, T.A. Mohanaprakash, K. Tongkachok, K. Kumar, Crime analysis using computer vision approach with machine learning, in: N. Marriwala, C. Tripathi, S. Jain, D. Kumar (Eds.), Mobile Radio Communications and 5G Networks, Lecture Notes in Networks and Systems, vol. 588, Springer, Singapore,2023, https://doi.org/10.1007/978-981-19-7982-8 25,

[5] Fei Xie, Ming Zhang, Jing Zhao, Jiquan Yang, Yijian Liu, Xinyue Yuan, A robust License Plate Detection and character recognition algorithm based on a combined feature extraction model and BPNN, J. Adv. Transport. Hindawi 2018 (2018) 1-14.

[6] Ibtissam Slimani, Abdelmoghit Zaarane, Wahban Al Okaishi, Issam Atouf, Abdellatif Hamdoun, An Automated License Plate Detection and Recognition System Based on Wavelet Decomposition and CNN Array vol. 8, Elsevier, 2020, pp. 1-7.

[7] K.B. Sathya, S. Vasuhi, V. Vaidehi, Perspective vehicle License Plate transformation using deep neural network on genesis of CPNet, in: Third International Conference on Computing and Network Communications vol. 171, Elsevier, Procedia Computer Science, 2020, pp. 1858-1867.

[8] Swati Jagtap, Analysis of feature extraction techniques for vehicle Number Plate Detection, Int. J. Comput. Sci. Inf. Technol. 6 (6) (2015) 5342-5346.

[9] K. Tejas, K. Ashok Reddy, D. Pradeep Reddy, M. Rajesh Kumar, Efficient License Plate Detection by Unique Edge Detection Algorithm and Smarter Interpretation through loT, 7th International Conference on Soft Computing and Problem Solving, 2020, pp. 1-11.

[10] Priyanka Prabhakar, P. Anupama, S.R. Resmi, Automatic vehicle Number Plate Detection and recognition, in: International Conference on Control, Instrumentation, Conunication and Computational Technologies (ICCICCT), 2014, pp. 185-190.

[11] Choong Young Jung, Keong Lee Kim, Tan Chye Cheah, License Plate number detection and recognition using simplified linear model, J. Crit. Rev. 7 (3) (2020) 55-60.

[12:07, 20/4/2025] Arun: [12] Olamilekan Shobayo, Ayobami Olajube, Ohere Nathan, Modupe Odusami. Obinna Okoyeigbo, Development of smart plate number recognition system for fast cars with web application, Appl. Computat. Intellig. Soft Comput. Hindawi 2020 (2020) 1-7.

[13] Weihong Wang, Jiaoyang Tu, Research on License Plate Recognition algorithms based on deep learning in complex environment, IEEE Access 8 (2020) 91661-91675.

[14] Palaiahnakote Shivakumara, CNN-RNN based method for license plate recognition, CAAI Trans. Intellig. Technol. 3 (3) (2018) 169-175.

[15] P. William, A. Shrivastava, P.S. Chauhan, M. Raja, S.B. Ojha, K. Kumar, Natural Language processing implementation for sentiment analysis on tweets, in: N. Marriwala, C. Tripathi, S. Jain, D. Kumar (Eds.), Mobile Radio Communications and 5G Networks, Lecture Notes in Networks and Systems,vol. 588,Springer,Singapore,2023

[16] S. Mishra, S. Choubey, A. Choubey, N. Yogeesh, J. Durga Prasad Rao, P. William, Data extraction approach using natural language processing for sentiment analysis, in: International Conference on Automation vol. 2022, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2022, pp. 970-972

https://doi.org/10.1109/ICACRS55517.2022.10029216..

[17] Deepak Narayan Paithankar, Abhijeet Rajendra Pabale, P. William Rushikesh Vilas Kolhe, Prashant Madhukar Yawalkar, Framework for implementing air quality monitoring system using LPWA-based IoT technique, Measurement: Sensors 100709 (2023),

https://doi.org/10.1016/j.mensen. 2023.100709, ISSN 2665-9174.

[18] K. Gupta, S. Choubey, Y. N, P. William, V. T.N, C.P. Kale, Implementation of motorist weariness detection system using a conventional object recognition technique, in: International Conference on Intelligent Data Communication Technologies and Internet of Things vol. 2023, IDCIoT), Bengaluru, India, 2023,pp.640-646, https://doi.org/10.1109/IDCT56793.2023.10052783.

[19] P. William, Y. N, V.M. Tidake, S. Sumit Gondkar, C. R, K. Vengatesan, Framework for implementation of personality inventory model on natural language processing with personality traits analysis, in: International Conference on Intelligent Data Communication Technologies and Internet of Things vol. 2023, IDCIoT), Bengaluru, India, 2023,pp. 625-628,https://doi.org/10.1109/IDCIoT56793.2023.10053501.

[20] Mark Sandler, Mobilenetv2, Inverted residuals and linear bottle necks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520.

[21] Kaiming He, Deep residual learning for image recognition", in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[22] P. William, A. Shrivastava, H. Chauhan, P. Nagpal, V.K. T. N, P. Singh, Framework for intelligent smart city deployment via artificial intelligence software networking, in: 3rd International Conference on Intelligent Engineering and Management vol. 2022, ICIEM), 2022, pp. 455-460, https://doi.org/10.1109/ICIEM54221.2022 9853119.

[21] Kaiming He, Deep residual learning for image recognition", in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[22] P. William, A. Shrivastava, H. Chauhan, P. Nagpal, V.K. T. N, P. Singh, Framework for intelligent smart city deployment via artificial intelligence software networking, in: 3rd International Conference on Intelligent Engineering and Management vol. 2022, ICIEM), 2022, pp. 455-460, https://doi.org/10.1109/ICIEM54221.2022 9853119.

[23] P. William, Y. N, S. Vimala, P. Gite, S.K. S, Blockchain technology for data privacy using contract mechanism for SG networks, in: 3rd International Conference on Intelligent Engineering and Management vol. 2022, ICIEM), 2022, pp. 461-465,

https://doi.org/10.1109/ICIEM54221.2022.9853118.

[24] S. Yuvaraj, A. Badholia, P. William, K. Vengatesan, R. Bibave, Speech recognition based robotic arm writing, in: V. Goyal, M. Gupta, S. Mirjalili, A. Trivedi (Eds.), Proceedings of International Conference on Communication and Artificial Intelligence, Lecture Notes in Networks and Systems, vol. 435, Springer,Singapore,2022

https://doi.org/10.1007/978-981-19-0976-4.3.

[25] S.S. Gondkar, D.B. Pardeshi, P. William, Innovative system for water level management using lot to prevent water wastage, in: International Conference on Applied Artificial Intelligence and Computing vol. 2022, ICAAIC), 2022, pp. 1555-1558,

https://doi.org/10.1109/ICAAIC53929.2022.9792746

[26] P. William, et al., Systematic approach for detection and assessment of dark web threat evolution, in: Romil Rawat, et al. (Eds.), Using Computational Intelligence for the Dark Web and Illicit Behavior Detection, IGI Global, 2022, pp. 230-256, https://doi.org/10.4018/978-1-6684-6444-1.ch013.

[27] R. Jadhav, A. Shaikh, M.A. Jawale, A.B. Pawar, P. William, System for identifying fake product using blockchain technology, in: 7th International Conference on Communication and Electronics Systems vol. 2022, ICCES, 2022, pp. 851-854, https://doi.org/10.1109/ICCES54183.2022 9835866.[28] P. William, N.K. Darwante, A.B. Pawar, M.A. Jawale, A. Verma, Framework for implementation of smart driver assistance system using augmented reality, in: N. Marriwala, C. Tripathi, S. Jain, D. Kumar (Eds.), Mobile Radio Communications and 5G Networks, Lecture Notes in Networks and Systems, vol. 588, Springer, Singapore, 2023, https://doi.org/10.1007/978-981-19-7982-8.30.