INTRODUCTION

Our goal is to create a smart system using advanced technology called Convolutional Neural Network (CNN) and Computer Vision. This system will keep an eye on vehicles in places like schools, residential areas and shopping complexes It's designed to monitor and manage vehicle access, ensuring a secure and organized environment.

In addition to this surveillance system maintain a database of authorised vehicles—which includes all the vehicles owned by the approved individuals—it will also monitor and record the entry and exit of vehicles.

The entrance and exit gates can be equipped with the proposed project. A network of security cameras installed there will be able to capture footage of cars coming and going from the property.

After tracking and identifying a vehicle, the system compares the licence plate information with the database to examine whether the vehicle is authorised.

By recording the entry and exit times of vehicles, our system allows for the calculation of the duration each vehicle spends on campus daily. Moreover, it effectively identifies any unfamiliar vehicles entering the premises. Through these capabilities, we not only strengthen security measures but also enhance our ability to monitor local events with increased efficiency.

LITERATURE REVIEW

The Paper [1] "Deep Learning based Real-time Stolen Vehicle Detection Model with Improved Precision and Reduced Look Up Time" by Dr. Shaik Shafi, T Pavan Sai Kumar Reddy, Rohan Silla and Musrath Yasmeen presents a study on the rising rate of vehicle theft in India, particularly in the region like Delhi and Bengaluru. This paper deals with the limitation of the existing methods and proposes a Deep Learning Based Real-Time Stolen Vehicle Detection Model using the advanced technologies like YOLO v 8, TensorFlow, and Open CV for the object recognition, license plate detection and image preprocessing, aiming to speed up the search process of the stolen vehicles. The main objective of the proposed model in this paper is to reduce the time required for vehicle detection in the CCTV footage.

For this purpose, the author uses the pretrained model called YOLO v8 model used for the real time Object detection. The author created a dataset with 400 images and the dataset is divided into training, validation and test partitions, and evaluates its performance of the model. Here this model detects the vehicle with an accuracy of 96%. This paper concludes by showing the results obtained from the proposed framework and providing insights into the effectiveness of real time vehicle detection.

The Paper [2] "Surveillance System using Moving Vehicle Number Plate Recognition" by Anurag Singh Rawat, Himanshu Devrani, Aman Yaduvanshi, Manvi Bohra, Indrajeet Kumar and Teekam Singh present an Automatic Vehicle Number Plate Recognition (ANPR) system using YOLO object detection and Optical Character Recognition (OCR).

The main aim of this proposed system is to automatically detect the vehicle number plates and extract the text from the number plates with high accuracy. There are several methodologies involves dataset acquisition, object detection using Yolo and the text recognition with preprocessing at the time of model evaluation. Evaluation shows that model efficiency with a 98.45% Precision and 96.06% Recall in the licence plate detection. The proposed system can perform the task for the detection of multiple vehicles simultaneously, searching a particular registration number for surveillance purpose and the metadata of vehicles to the database. This paper also talks about the significance of enhancing surveillance and traffic rule enforcement in various environments.

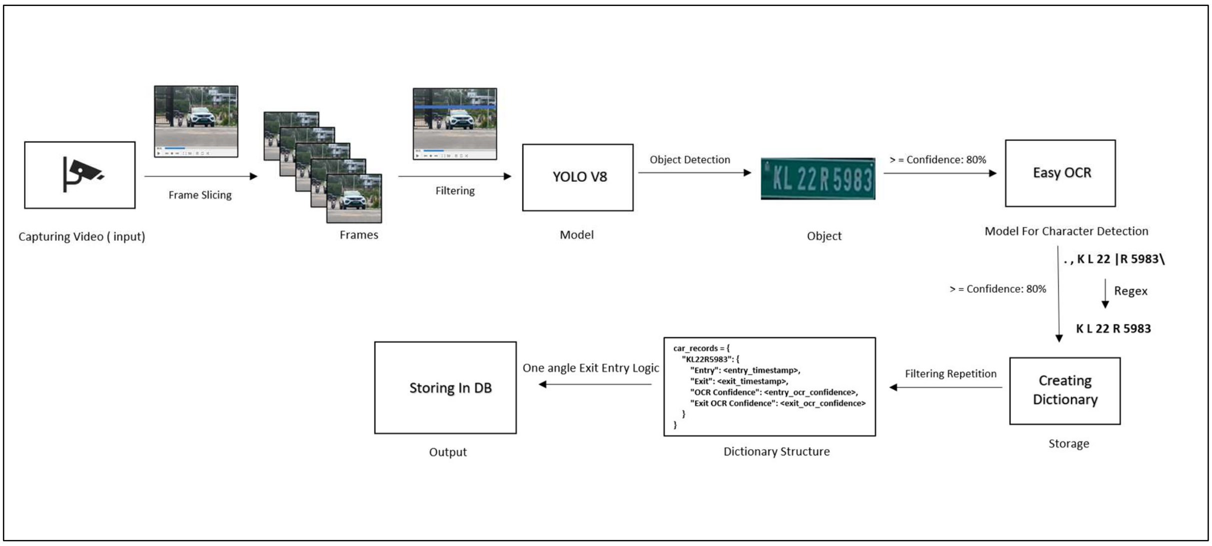
PURPOSE OF THE PROJECT

The primary purpose of our Automated Vehicle Entry and Exit Tracking System project is to develop a sophisticated, reliable, and automated mechanism for monitoring and recording the entry and exit times of vehicles within our campus environment and residential areas. Our advanced system that utilizes computer vision and deep learning techniques, specifically Convolutional Neural networks for image analysis and the easy OCR for text extraction from the numberplates, to track the entry and exit of each vehicle by jointly analysing textual content and associated images.

This project aims to create a comprehensive solution capable of automating the entry and exit logging process will streamline operations, reduce wait times at entry and exit points, and minimize the need for manual data entry and verification. With the help of real-time video capturing and processing, the system ensures that the information collected is accurate and immediately available for review, which is essential for effective traffic management and planning. By achieving these objectives, the project aims to set a new standard in vehicle management within institutional settings, ensuring safety, effective traffic management and planning to future technological advancements.

METHODOLOGY

System Architecture



system architecture

We use a camera at the university entrance, recording car videos at 60 frames per second. To make things efficient, we filter out faraway cars, keeping the ones closer for analysis. Then, we apply the YOLO V8 model for object detection. This method helps us better understand the traffic flow around the campus.

After detecting objects, we only consider models with a confidence level of 80% or higher. We then pass the results to a character detection model, Easy OCR, and we also filter the results to include only those with a confidence level of 80% or higher. The obtained results may contain anomalies, and we address them by filtering using regular expressions. We also structure the numbers using regular expressions and after that we create a dictionary with the following format. Then we apply a one-hand logic. Specifically, we establish an ideal time, detecting vehicles only after that time. Finally, we store the processed information in a database.

Data Collection

We have a dataset comprising approximately 1000 images, with 500 images acquired through manual photography of vehicles. Additionally, we obtained two datasets from Kaggle, one consisting of 100 images in PNG format and another with 230 images in JPG format. The remaining images were manually extracted from websites such as OLX.

We selected a total of 609 images, allocating 57 for validation and 30 for testing purposes. While collecting data we ensure consistency in image quality across the dataset to avoid biases in model training. Also, we ensure that the dataset include variation in Indian license plate in terms of format, font and regional differences.

Data Preprocessing

Data preprocessing and augmentations is important for ensuring the integrity of image and suitable for further analysis. At first, the augmentation of images was done by applying auto-oriented and resizing all the images to a standardized dimension of 640x640 pixels. Auto-oriented image will help for consistent analysis and model training. Resizing to a standard 640x640 resolution maintains consistency in input dimensions, Important for model compatibility and efficient processing. The “stretch” method is used for resizing and it will maintain the original aspect ratio of the images. Augmentation process ensure that all the images are correctly oriented. This approach is crucial as it prevents distortion and maintains essential features within the images.

The model becomes more versatile in object detection when it is zoomed via cropping (0–20%), which allows the model to understand objects at different scales. Rotation augmentation (-15° to +15°) improves the dataset by exposing the model to objects from different angles. Grayscale additions (5%), saturation (-46% to +46%), and brightness adjustments (-25% to +25%) show the model to a wider range of colour variations and lighting conditions. These techniques allow for the use of diverse colour representations. To train the model to handle such scenarios effectively, minimal blur (up to 1px) and noise within bounding boxes (up to 1% of pixels) are used. Bounding box adjustments that include 90° clockwise and anticlockwise rotations as well as upside-down orientations improve the model's ability to identify objects in a variety of orientations. The diversity of the dataset is considerably increased by these preprocessing and augmentation steps.

Model Architecture

The YOLOv8 model architecture is the latest version in the You Only Look Series. This model is designed for accurate real-time object detection and image segmentation. It employs a single convolutional neural network (CNN) that processes an entire image in one evaluation, predict both bounding boxes and class probabilities simultaneously. The Yolo v8 model divide the input images into grid, with each cell responsible for predicting bounding boxes and confidence scores indicating the presence of an object and predict the accuracy of the box. In addition to that it also predicts the class probabilities for each box and determine the category of detected objects. The optimization techniques like batch normalization, data augmentation and transfer learning are used to improve the model performance. YOLOv8 stands out for its balance of speed and precision, making it well-suited for applications requiring fast and reliable object detection.

Easy OCR

Easy OCR stands for "Optical Character Recognition. It helps the quick conversion and extraction of text from images. It is a python module for extracting text from image. Over 80 Languages are Supported. It is functional with Django, Flask, and other Python frameworks and can be installed as a Python package. The output will be in a list format containing Text Content, Coordinates, Font Information and Confidence Score.

Text Content refers to the recognized text from the document. In this context, Coordinates denote the position or bounding box of each text block within the document, and Font Information encompasses details about the font, such as its size, style, and colour. The Confidence Score serves as a gauge of the system's confidence in the accuracy of text recognition.

Easy OCR utilizes deep learning and computer vision technique to recognize text from images. It uses pre-trained computer models to recognize and understand characters with in the given images. It Processes images to recognize and extract text, providing the output in a readable format. CRAFT functions as the detection algorithm, while CRNN operates as the recognition model. Key elements of CRNN encompass feature extraction, sequence labelling using LSTM, and decoding through CTC. CRAFT employs character-level segmentation, generating bounding boxes that delineate specific characters or text regions within the image. CRAFT segments data at the character level. It provides a precise and detailed depiction of the text layout by identifying regions that correspond to individual characters.

CRNN (Convolutional Recurrent Neural Network) is a kind of neural network architecture designed for sequence recognition applications like OCR.

In CRNN, Feature Extraction involves the initial stage using Convolutional Layer to extract essential elements from the provided image. Sequence Labelling, implemented with LSTM (Long Short-Term Memory), is employed for labelling sequences. Decoding in this context refers to training Sequence-to-sequence models with CTC, eliminating the requirement for aligning input and output sequences. This approach is commonly utilized in OCR tasks to interpret the character sequence predicted by the network.

Open CV

OpenCV supplies various functions and tools for analysing images and videos, serving as a valuable asset for computer vision projects. Features:

OpenCV comes with diverse features, offering functions for analysing images and videos, along with tools for basic image adjustments and advanced machine learning algorithms.

In our project, we utilize OpenCV to process and manipulate video feeds. This is done to achieve optimal performance in both object detection and Optical Character Recognition (OCR).

Working of the Code

The code begins by loading the YOLO model for object detection and the Easy OCR (Optical Character Recognition) reader for reading the text, to detect the vehicle and their number plates. Then the code captures video frames from a specified path in the code and processes them in real time and our next step is to set up a virtual line across the video frame to detect vehicle crossings. For each video frame, YOLO model identifies vehicles based on the confidence level and detected vehicles are marked with bounding boxes. When a Vehicle crosses the virtual line, its number plate is extracted and pre-processed for OCR. After preprocessing, Easy OCR then reads the number plate's text. The text from OCR is cleaned and validated against a regular expression pattern to ensure it matches the standard number plate format. Valid number plates and their corresponding entry/exit timestamps are recorded. The processed video after the model evaluation phase and the detection results are displayed in real time. The program can be excited by pressing the ‘q’ key. Finally, all recorded vehicle entries and exits are saved into an Excel file for further analysis.

Model Summary

We are using the pre- trained model named YOLO (You Only Look Once) model. First, we collect the images of different vehicles for training purposes. Then we manually annotated the bounding boxes for each image and distributed the vehicle images in train, test and validation parts. We have a dataset containing 609 images for training purposes, 57 images for the validation process and 30 images for testing the model. Then we are applying the Image preprocessing ang augmentation steps to maintain the dataset diversity and it will be boosting the object detection accuracy and the reliability. The image preprocessing steps include Auto-Oriented and Resize. Auto-Oriented adjustment corrects the image orientation inconsistencies. Resizing the images to a standardized 640x640 resolution maintains consistency in the input data dimensions and for efficient processing.

The Augmentation steps includes Horizontal flipping, Crop, Rotation, Gray Scale, saturation, Brightness, Blur, Bounding Box Modifications and the Noise in Bounding Box. All these augmentation processes have their own importance for improving the accuracy of object detection and reliability.

After the data preprocessing and data augmentations then we must train our model using these clean data. Our model comprises of 168 layers with a total of 11,125,971 parameters. During the training phase, the model completed 150 epochs in approximately 0.645 hours. The stripped optimizer files for both the last and weights are 22.5 MB each.

The Model's performance on validation showcases impressive results. It processed 57 images in the validation stage and detected a total of 59 instances across classes. Instances represent the total number of objects or instances detected across all images. The Precision metrics for the bounding box prediction is 93.2% and recall is 96%. The mean Average Precision (mAP50) stands at 95%.

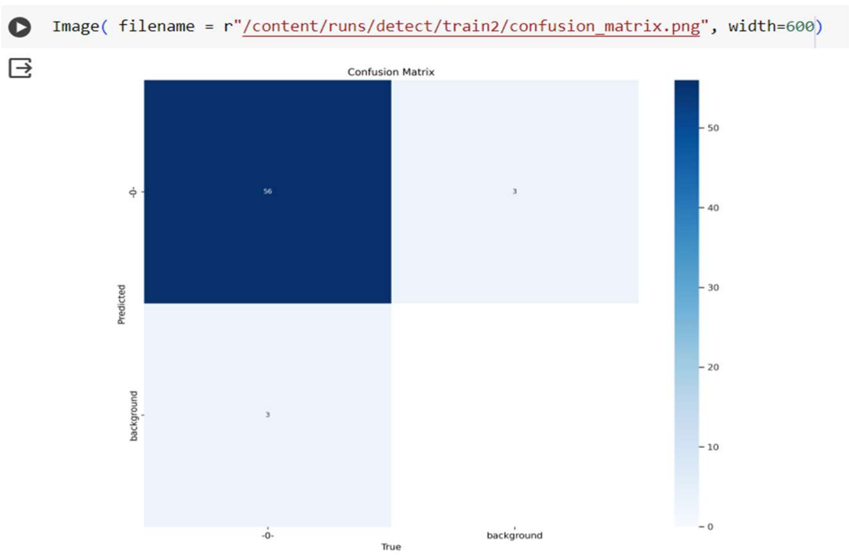
The overall Map50-95(mean Average Precision from 50% to 95% confidence) is about 68.6%.

mAP50: Mean Average Precision at IoU (Intersection over Union) threshold of 0.5. This metric evaluates the precision of the model at different IoU thresholds (here, at 0.5) and calculates the average. A higher mAP50 indicates better precision at identifying objects.

mAP50-95: Mean Average Precision across IoU thresholds from 0.5 to 0.95. This metric computes the average precision over a range of IoU thresholds from 0.5 to 0.95, giving an overall performance evaluation across various levels of bounding box overlap.

Moreover, in terms of speed, the model exhibits efficiency, with preprocessing taking 0.2ms per image. Preprocessing time represents the time taken by the model to prepare or preprocess each image before it is fed into the neural network for inference. Preprocessing tasks might include resizing, normalization, or any transformations required to adapt the input image to the model's requirements. Then the Inference time of our model is 5.3ms per image. Inference time is the measurement of time taken by the model to perform its computations and provide output (e.g. Identifying objects in the image) based on the input data (e.g. .an image). The loss computation time is reported as 0.0ms per image which may suggest that this information might not be relevant or collected during inference. The Post processing of our model is 2.4ms per image, it represents the time taken after the model's inference to refine or process the predictions. Post processing involves tasks like filtering bounding boxes based on confidence scores and non-maximum suppression (removing overlapping boxes). These time measurements provide insights into the computational performance of the model at different stages of the inference pipeline, highlighting where most of the time is spent during the object detection process.

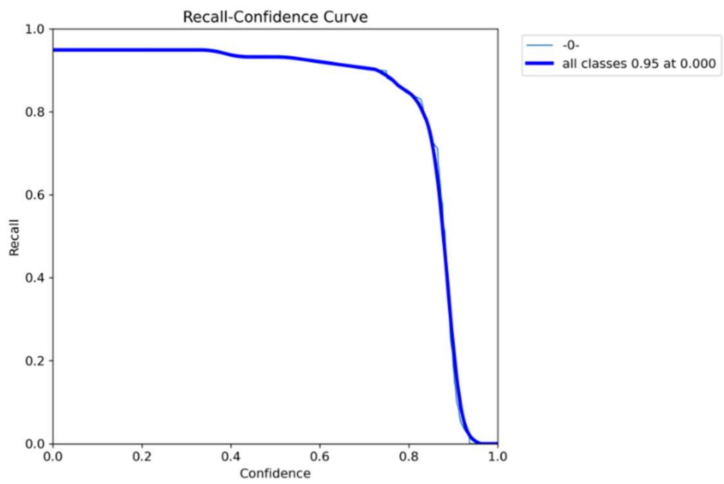
The proposed model is initially trained on 609 images and validated on a collection of 57images. After validation, 30 external images were used the testing purposes. The performance of the model is evaluated in the form of training loss, training accuracy, validation loss and validation accuracy. The obtained confusion matrix of the model is shown below,



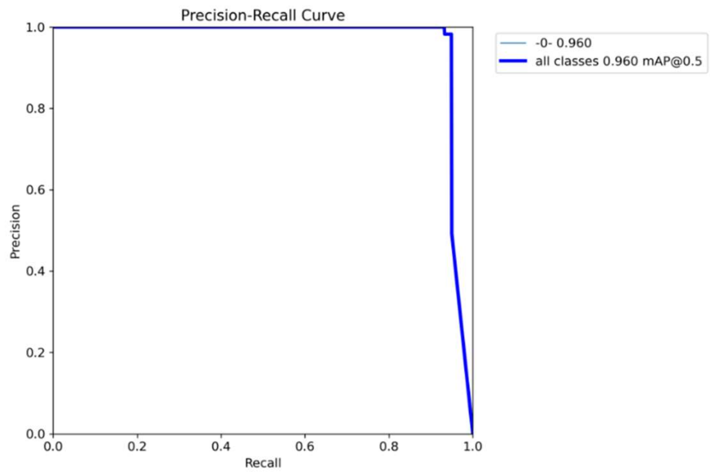
Confusion Matrix

A confusion matrix is a crucial tool for assessing a machine learning model's accuracy for each class. The matrix has two axes, "True" on the horizontal axis and "Predicted" on the vertical axis. "True" refers to the actual class labels, while "Predicted" refers to the labels predicted by the classification model. There seem to be two classes, which are "0" and "background." one class to represent the positive class (e.g., presence of an object) and another to represent the negative class (e.g., absence of an object or just the background). The top-left cell (dark blue) with the number 56 indicates true positives (TP) or true negatives (TN), depending on which class is considered positive. It represents the number of times the model correctly predicted class "0." The bottom-right cell (light blue) with the number 3 represents the true negatives (TN) or true positives (TP), again depending on class designation, showing the instances where the model correctly identified the "background”. The top-right and bottom-left cells are not visible due to the colour scheme, but they would typically show the number of false positives (FP) and false negatives (FN), respectively. These are the instances where the model incorrectly predicted the class. By using the obtained confusion matrix, precision, recall is calculated. Moreover, it offers details on other crucial performance measures including F1score, recall, and accuracy. All these measures should be considered when evaluating a model's performance.

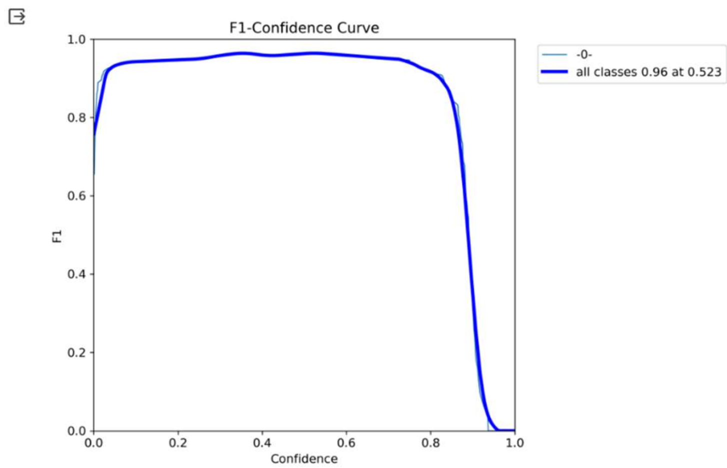
The obtained recall vs confidence curve is shown below, it demonstrates how, as the amount of confidence rises, the recall of the model's predictions varies. This curve may be used to assess the model's effectiveness and choose a confidence level that strikes a balance between strong recall and adequate accuracy.



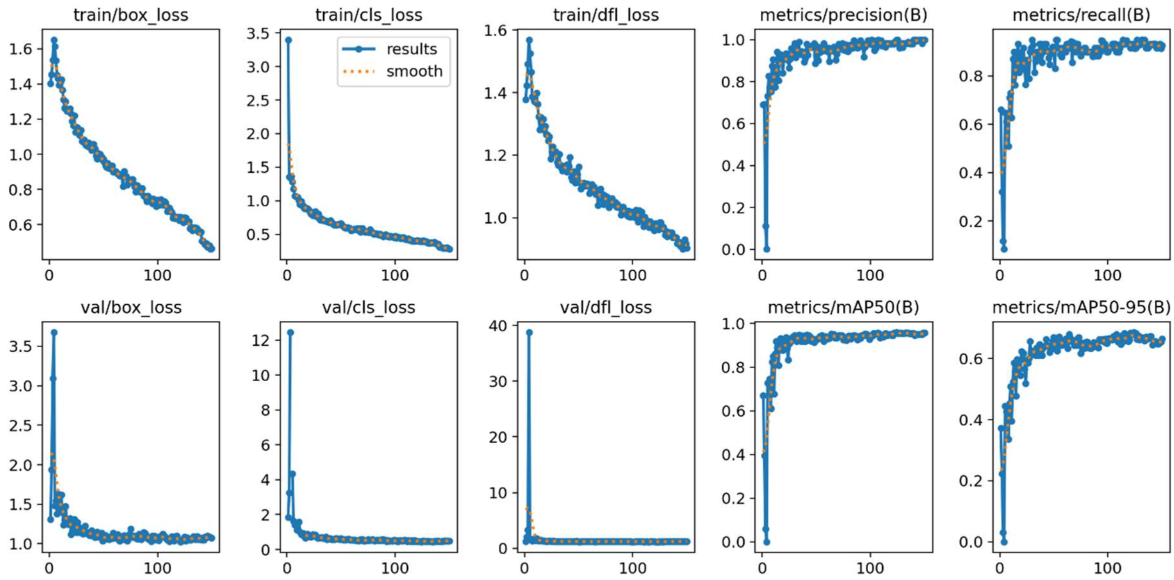
When it comes to assessing a model's performance where both lowering false positives and lowering false negatives are crucial, the accuracy vs recall graph is helpful. The obtained precision VS recall curve is shown below,



F1 confidence curve analysis how changes in the confidence threshold affect the F1 score.By analysing the curve, we can understand the trade-off between precision and recall. The given graph shows whether increasing the confidence threshold leads to higher precision but lower recall or vice versa. The obtained F1 confidence curve of our model is shown below,



The obtained results of YOLO model in terms of train/box loss, train/cls_loss, Val/box loss, Val/cls_loss, train/dfl_loss, precision, metrics/recall(B), metrics/mAP50(B), metrics/mAP50-95(B)is shown below,



The proposed model was tested against a validation batch consisting of images. The proposed model is performing quite well in different scenarios.

we achieved an accuracy of roughly 96% for static photos. We obtained an accuracy of 68.7% after testing the model using real-time footage.

RESULT AND DISCUSSION

The initial priority is about how accurately and quickly the proposed model detects the vehicles. The pre-trained YOLO v8 is employed for license plate detection and Easy OCR is used for text extraction from the number plates. The proposed model YOLO v8 accurately spots the vehicles in various frames and is highly efficient for real-time object detection. The Easy OCR reader performs well in reading text from number plates, but its effectiveness can depend on several factors like image clarity and lighting conditions.

The proposed system has effectively performed in real- time situations and processed video frames for consistently identifying vehicles. This implies that the proposed system is suitable for use in continuous and immediate monitoring. Even so, it is important to note that the speed and efficiency of processing may be influenced by the hardware in use and there some challenge may face in several scenarios with high traffic or high-resolution video feeds.

The system ability to track the entry and exit of vehicles using a virtual line in the video frame was accurate. It consistently recorded timestamps whenever vehicles crossed this virtual line, especially their entry or exit. This feature of the proposed system is more advantageous for organizing vehicles traffic in restricted settings, such as educational campuses.

The use of regular expression-based methods to filter OCR results ensured that only relevant and correctly formatted number plate data was processed. This approach played a crucial role in minimizing errors and enhancing the overall reliability of the proposed system. The overall performance of the proposed system is excellent, but it has faced some challenges. For example, it may struggle with reading licence plates in bag lighting condition or when the visibility is poor and its speed is dependent on the computer it is running on, which could be a problem in places with limited resources.

From the user's perspective, the system is easy to use. It automatically finds vehicles and records their details without needing manual input. The results are shown in an Excel sheet, which is practical for administrative tasks. However, the current interface is basic and more focused on functionality than creating an engaging user experience.

Results

A	B	C	D	E
	Entry	**OCR Confidence**	**Exit**	**Exit OCR Confidence**
KL01BU788	2024-01-05 14:50:06	0.999209021	2024-01-05 14:51:07	0.993738327
KL22R5983	2024-01-05 14:53:41	0.918315867	2024-01-05 14:55:11	0.982710149

Output Excel

FUTURE SCOPE

- We plan to implement a system that alerts security officers when an unauthorized vehicle is detected. This involves integrating a Raspberry Pi microprocessor and utilizing a Pi Cam camera module to develop a compact, portable device.
- We want to make a website that's easy to use and can quickly find out who owns a vehicle using APIs. This will help simplify the identification process, making it more user-friendly and efficient for everyone.
- In the next phase of our project, we plan to create a single database to save space and efficiently organize surveillance video recordings. This centralized system will streamline storage management and making it more effective and space-saving.

CONCLUSION

The project successfully shows how advanced technologies like object detection and optical character recognition can be combined to automate the tracking of vehicles entering and exiting out of a place. The system effectively records and processes vehicle movements instantly, offering a secure and dependable way of tracking that is better than the traditional manual logging methods.

This system will help us to increase the roll of security.one angle exit-entry logic system implemented on this project will help us reduce the complexity in operation. In future system can be enhance to more resourceful manner.

REFERENCE

I. Research paper [1] “Deep Learning based Real-time Stolen Vehicle Detection Model with Improved Precision and Reduced Look Up Time” by Dr. Shaik Shafi, T Pavan Sai Kumar Reddy, Rohan Silla and Musrath Yasmeen. https://ieeexplore.ieee.org/document/10205684

II. Research paper [2] “Surveillance System using Moving Vehicle Number Plate Recognition” by Anurag Singh Rawat, Himanshu Devrani, Aman Yaduvanshi, Manvi Bohra, Indrajeet Kumar and Teekam Singh. https://ieeexplore.ieee.org/document/10212426

III. YOLO (You Only Look Once) Pre-trained model Documentation. https://docs.ultralytics.com/modes/track/

IV. Easy OCR documentation https://www.techtarget.com/searchcontentmanagement/definition/OCR-optical-character-recognition