American Journal of Electronics & Communication

Shaik Nehar et. al., American Journal of Electronics & Communication, Vol. IV (3), 16-23

Automatic Number Plate Recognition

Shaik Nehar
Department of AI&DS ,Bachelor of Scholars
Vasireddy venkatadri institute of technology
Nambur,guntur,522007.
neharimran@gmail.com

Sanagala Teja
Department of AI&DS ,Bachelor of Scholars
Vasireddy venkatadri institute of technology
Nambur,guntur,522007.
sanagalateja12@gmail.com

Lohith vattikuti
Department of AI&DS ,Bachelor of Scholars
Vasireddy venkatadri institute of technology
Nambur,guntur,522007.
lohithvchowdary@gmail.com

Sai vignesh chintala
Department of AI&DS ,Bachelor of Scholars
Vasireddy venkatadri institute of technology
Nambur,guntur,522007.
Ch.vignesh2003@gmail.com

SrinivasaRao Tummalapalli
Department of AI&DS ,Bachelor of Scholars
Vasireddy venkatadri institute of technology
Nambur,guntur,522007.
srinut422@gmail.com

Mahendra Eda
Department of AI&DS ,Bachelor of Scholars
Vasireddy venkatadri institute of technology
Nambur,guntur,522007.
mahindramahi08@gmail.com

Abstract:-** This innovative and comprehensive approach harnesses the combined power of YOLOv8 and EasyOCR technologies for Automatic Number Plate Recognition (ANPR). The primary objective is to extract valuable data from vehicle images, even when they are blurred or distorted, and utilize this data for further analysis using machine learning models.The cornerstone of this approach is the integration of two cutting-edge technologies: You Only Look Once (YOLO) V8 for Region of Interest (ROI) detection and EasyOCR for optical character recognition. The workflow begins with the utilization of YOLOv8, which excels in identifying and correcting distorted license plates within a single image. Its initial task is to pinpoint the Region of Interest (ROI). Once the ROI is identified, a series of pre-processing steps are employed to enhance the image quality before feeding it into the EasyOCR model for character recognition.Several significant advantages distinguish this approach. Firstly, it possesses the capability to deblur images, enabling accurate data extraction even from blurred or challenging images. Secondly, it synergizes the strengths of YOLOv8 for precise ROI detection and EasyOCR for accurate character recognition, resulting in a robust and dependable ANPR system. Lastly, the system is optimized for real-time processing, rendering it suitable for applications requiring swift and efficient vehicle monitoring.The key innovation lies in the ability to extend its functionality to video data, allowing for the detection of number plates in dynamic settings. It not only detects license plates but also provides an annotated video with comprehensive license plate details. This holistic approach facilitates a more complete understanding of vehicular movements and activities.By harnessing the capabilities of YOLOv8 and EasyOCR technologies, this approach delivers a comprehensive and sophisticated solution for Automatic Number Plate Recognition. It enhances accuracy, efficiency, and the ability to handle challenging scenarios, such as blurred images and dynamic video feeds. This innovative system serves as a valuable tool for promoting safe, secure, and modern transportation systems.

***keywords:** Yolov8,EasyOCR,Region of Interest, Character Recognition*

I. Introduction

Automatic Number Plate Recognition (ANPR) is a technology that optimizes the movement of automobiles over transport networks. ANPR involves acquiring and analyzing images from traffic surveillance cameras, and it has gained momentum in recent years due to the advancements in neural networks and deep learning[2]. ANPR can be applied to many areas, like traffic law enforcement, automatic toll tax collection, car parking systems, and automatic vehicle parking systems[6].

The steps involved in ANPR are image acquisition, preprocessing of the image, finding the region of interest (ROI), segmentation, and optical character recognition. The initial phase of ANPR is image acquisition, where input images can be extracted from traffic surveillance videos. The second step is finding the Region of Interest, which in this case is a license plate present in the image. Edge detection is the most common method to use for number plate detection, and more techniques are used for plate detection[2]. In the next stage, after the detection of the plate, segmentation is done to identify the regions where alphanumeric characters are present. The final step is to recognize the segmented region as alphanumeric characters.

To improve the accuracy and efficiency of ANPR, researchers have proposed a novel approach that combines the power of YOLOv8 and EasyOCR technologies[1].While EasyOCR can identify the characters on a license plate that has been identified, the YOLOv8 model can detect and correct numerous distorted license plates in a single image. The suggested method has a number of benefits, including the capacity to handle blurry photos, which makes it an important tool for contemporary, secure, and safe transportation systems[4].

The ANPR systems placed along the roadways can be used to detect stolen automobiles in an effective manner. This paper presents a recognition method that uses the YOLO algorithm for Automatic Number Plate Recognition

(ANPR)[1].A Convolutional Neural Network (CNN) was suggested in another study to be capable of identifying and correcting several deformed license plates in a single image, which would then be fed into an optical character recognition (OCR) approach to get the desired outcome[2]. The Plate Recognizer team has created Automatic License Plate Recognition (ALPR) software that is location-specific and functional in any setting.

The final segment is as follows: Section II provides a summary of the literature review of the previous study. The techniques employed for the planned study are described in Section III. The outcomes of the chosen strategy are explained in Section IV. The suggested work is concluded in Section V, which also makes mention of future work.

II. Related Work

Finding and identifying license plates in photographs is the duty of ANPR[2-3]. Character segmentation, vehicle detection, license plate detection, and character recognition are the four subtasks that typically make up a sequential pipeline.We'll just call the culmination of the past two tasks optical character recognition for short.

License plate localization is an essential step in Automatic Number Plate Recognition (ANPR), and traditional methods based on a priori information are generally classified as color texture, shape regression, and edge detection[5]. However, these methods have limitations because they rely on manual feature extraction, which is not well-suited to the diversity of images. Target identification techniques based on deep learning have advanced quickly in recent years, and the algorithms can be broadly split into two types.The first category generates a part of the candidate region by the algorithm, and then the candidate region is classified and positioned again. End-to-end detection techniques fall under the second group; these algorithms immediately obtain the target's coordinates and class probability. ANPR systems that use deep learning algorithms have shown high accuracy and efficiency. Ibtissam Slimani et al based their license plate detection on wavelet transform, followed by validation of potential regions using a CNN classifier. The YOLOv8 algorithm is an example of an end-to-end detection algorithm that is widely used in ANPR systems. It directly gets the location coordinates and class probability of the target, making it highly accurate and efficient[6].

Automatic Number Plate Recognition (ANPR) is a widely used computer vision application that involves finding and recognizing license plates in images. In the license plate recognition stage, traditional recognition algorithms segment the license plate characters one by one and then use optical character recognition (OCR) technology to recognize each character. However, this method has poor recognition efficiency[3-5]. Many ANPR systems can only achieve good recognition under specific conditions, such as good weather conditions, adequate lighting, fixed scenes, and facilities.It is still difficult to recognize license plates in complex situations due to problems including poor nighttime lighting, rain, snow, and covered or blurred license plates.In recent years, deep learning-based target detection methods have developed rapidly, and the algorithms are mainly divided into two categories. One category generates a part of the candidate region by the algorithm, and then the candidate region is classified and positioned again. End-to-end detection techniques fall under a different category and directly obtain the target's coordinates and class probability[6]. Our ANPR system uses an end-to-end method based on deep learning that optimizes the efficiency and accuracy of recognition. The system uses YOLOv8, a deep convolutional neural network, for license plate detection, and EasyOCR for character segmentation and recognition[3]. The combination of these two tools forms a sequential pipeline for ANPR, which consists of the four subtasks mentioned above. The OCR technology used in EasyOCR is robust and has very high accuracy, which is essential for accurate character recognition. Our ANPR system is capable of detecting license plates in unconstrained scenarios, which means that it can handle distorted text and high font variability[1].

According to a review of the literature, the following restrictions apply to the suggested algorithms:

1. Low image quality
2. Less Precision
3. Dim illumination and weak contrast
4. Increased Cost of Computation
5. A lack of regulations for automobile license plates
6. Characters that have been improperly segmented won't be recognized.

III. Methodology

My methodology involves the usage of YOLOv8 and EasyOCR for Automatic License Plate Recognition (ALPR). YOLOv8 is a deep convolutional neural network that is used for vehicle recognition and license plate detection, while EasyOCR is used for character segmentation and recognition. The combination of these two tools forms a sequential pipeline for ALPR.Fig. 1 shows the flow we followed.

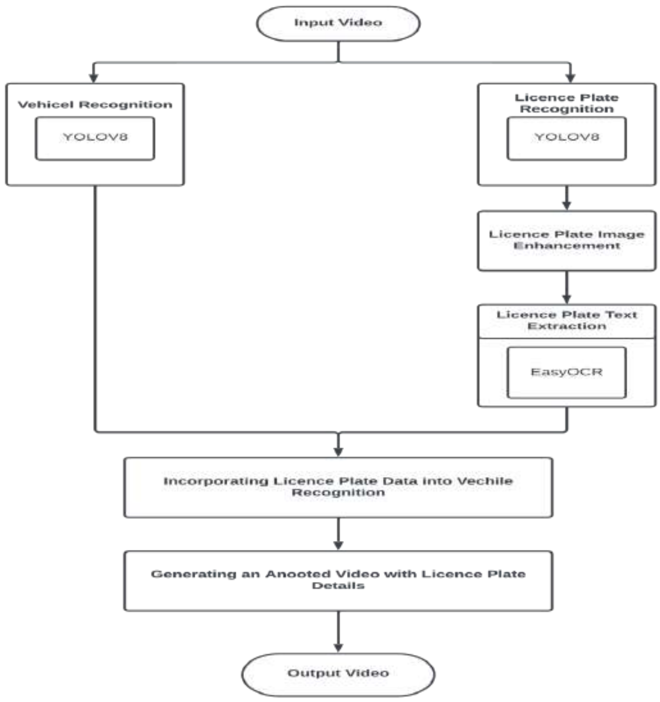


Fig1: Flow diagram prepared for proposed methodology

A. 3.1 Vechile Recognition:

Vehicle recognition is a crucial component of modern computer vision systems, with applications ranging from traffic management to surveillance and autonomous vehicles. In this context, the YOLOv8 algorithm plays a pivotal role as a powerful and efficient object detection framework[4]. Trained on the extensive COCO dataset, YOLOv8 exhibits the capability to detect a wide range of objects, including vehicles, in real-time video streams.

Fig:Yolov8 Architechture

The process of vehicle recognition begins with the input video stream, which is sequentially processed frame by frame[4-5]. Each frame is analyzed by the YOLOv8 model, which identifies potential vehicles within the image. The algorithm returns bounding boxes around these detected vehicles, accompanied by confidence scores that reflect the model's confidence in its predictions.

To ensure that only vehicles are considered for further analysis, the detected bounding boxes are filtered based on their associated class identifiers[4]. Vehicles typically have specific class identifiers, making it possible to distinguish them from other objects that may be present in the scene.

The resulting set of filtered bounding boxes, representing vehicles in the frame, forms the basis for subsequent analysis[5]. These bounding boxes are then passed to the license plate recognition system, which focuses on the regions of interest (ROI) containing the license plates of the detected vehicles. This two-step process not only identifies vehicles within the video stream but also paves the way for detailed analysis of license plate information, such as recognition and extraction[5].

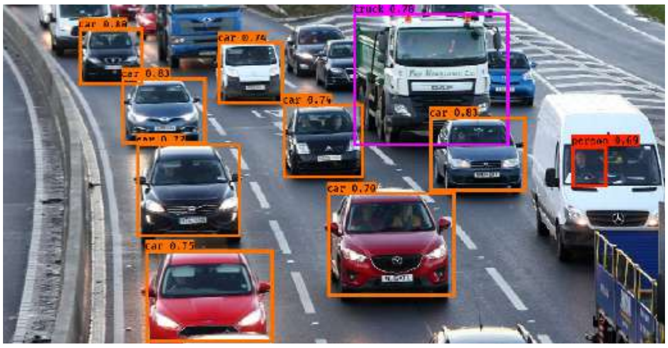


Fig:Vechile Recognition

B. 3.2Licence Plate Recognition:

1) Preparing the dataset

The workflow begins with the installation of the Roboflow library, a tool that streamlines data management and preprocessing for machine learning projects, including LPR. The library facilitates the handling of image datasets, making it easier to prepare the data for training[6].

Within this context, a specific project and dataset are accessed using the Roboflow API. The chosen dataset likely contains a collection of images with labeled license plates, which serves as the training data for the LPR model. By leveraging this data, the system can learn to recognize and interpret license plates accurately3].

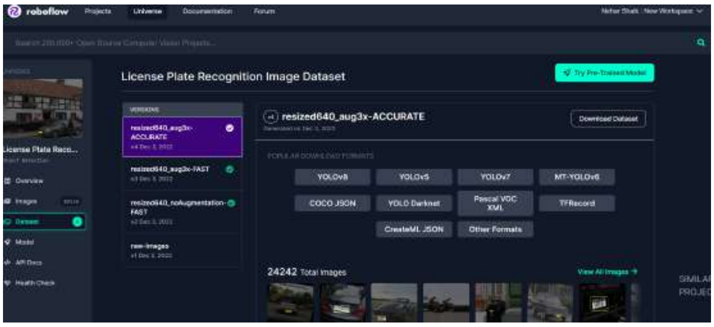
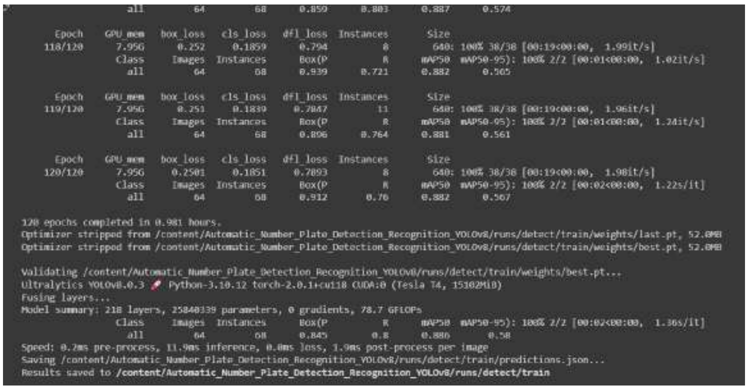


Fig:Data form Roboflow

2) Train the Model

In our project, we utilized YOLOv8 as our chosen model and conducted training over 120 epochs, completing the process in a notably reduced time of 0.981 hours. This runtime is significantly faster compared to the YOLOv5 model discussed in Figure , in collaboration with Google Colab. Additionally, in Figure of our study, we present the outcomes of YOLOv8's training on the training dataset, showcasing the recognized labels, as well as providing precision, recall, and mAP (mean Average Precision) values. This performance assessment highlights the effectiveness of our chosen YOLOv8 model in object detection tasks.



```
                 all         64         68      0.859      0.803      0.887      0.574

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    118/120      7.95G      0.252     0.1859      0.794          8        640: 100% 38/38 [00:19<00:00,  1.99it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.02it/s]
                   all         64         68      0.939      0.721      0.882      0.565

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    119/120      7.95G      0.251     0.1839     0.7847         11        640: 100% 38/38 [00:19<00:00,  1.96it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.24it/s]
                   all         64         68      0.896      0.764      0.881      0.561

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    120/120      7.95G     0.2501     0.1851     0.7893          8        640: 100% 38/38 [00:19<00:00,  1.98it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.22s/it]
                   all         64         68      0.912       0.76      0.882      0.567

120 epochs completed in 0.981 hours.
Optimizer stripped from /content/Automatic_Number_Plate_Detection_Recognition_YOLOv8/runs/detect/train/weights/last.pt, 52.0MB
Optimizer stripped from /content/Automatic_Number_Plate_Detection_Recognition_YOLOv8/runs/detect/train/weights/best.pt, 52.0MB

Validating /content/Automatic_Number_Plate_Detection_Recognition_YOLOv8/runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.3 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Fusing layers...
Model summary: 218 layers, 25840339 parameters, 0 gradients, 78.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.36s/it]
                   all         64         68      0.845        0.8      0.886       0.58
Speed: 0.2ms pre-process, 11.9ms inference, 0.0ms loss, 1.9ms post-process per image
Saving /content/Automatic_Number_Plate_Detection_Recognition_YOLOv8/runs/detect/train/predictions.json...
Results saved to /content/Automatic_Number_Plate_Detection_Recognition_YOLOv8/runs/detect/train
```

Fig:Yolov8 Model Training

3) Evaluating the model performance

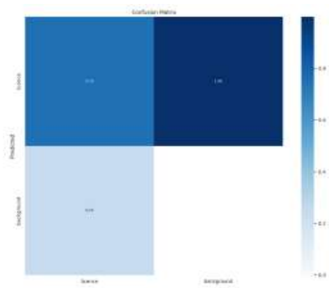


Fig:Confusion Matrix Of Model

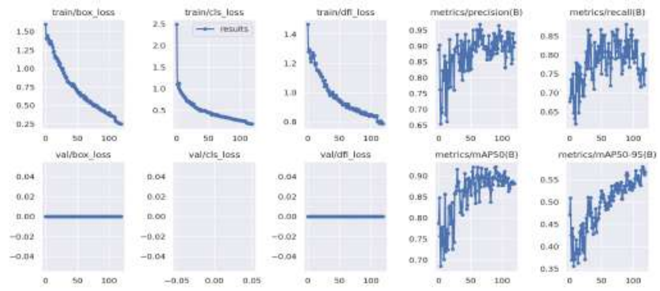


Fig:Metrix on object identification after testing the data

4) Checking the output generated from the model

Fig:Plates identified by Yolov8

It's important to note that training a license plate recognition model is a resource-intensive task that requires a substantial amount of labeled data, computational power, and time. The choice of architecture (YOLOv8 in this case) is significant, as it influences the model's performance in terms of accuracy and speed[5].

C. *3.3Licence Plate Image Enhancement*

Within the realm of LPR, a crucial step involves the initial processing of license plate images to enable precise character identification[2]. The provided segment of the process emphasizes the significance of this preparatory phase, which entails transforming license plate images into grayscale and then applying thresholding[2-3].

Fig: Original Licence Plate

The conversion to grayscale simplifies the license plate image by eliminating color information, resulting in a single-channel image where pixel values represent varying degrees of brightness[2]. This simplification reduces the intricacy of the image data, streamlining subsequent processing steps to concentrate exclusively on luminance data. Grayscale images prove particularly valuable for character recognition, as they remove any potential impact from color variations that may be present.

Fig:license_plate_crop_gray

Following the grayscale conversion, the technique of thresholding is applied. This process involves converting the grayscale image into a binary format, where pixel values are categorized as either black or white based on a predetermined threshold value[2]. In this instance, a threshold value of 64 is employed. Pixels with values equal to or exceeding 64 are rendered as black (0), while those below this threshold are depicted as white (255). The utilization of the "THRESH_BINARY_INV" flag signifies the application of inversion, effectively swapping the foreground and background colors.

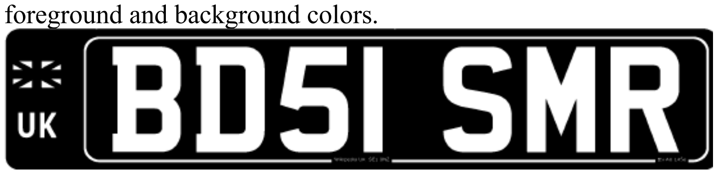


Fig:license_plate_crop_thresh

The significance of this thresholding procedure lies in its role in separating characters on the license plate from the background[1]. Through this transformation into a binary format, the characters usually become black against a white backdrop, resulting in heightened contrast and improved visibility for subsequent optical character recognition (OCR) techniques.

Within the broader context of license plate recognition, this preprocessing stage holds paramount importance for enhancing overall accuracy and dependability. It readies the license plate image for character segmentation and, subsequently, character recognition. This preparation ensures that OCR algorithms can proficiently detect and interpret the characters displayed on the license plate.

D. **3.4 Licence Plate Text Extraction**

Text extraction from license plates is a critical component of license plate recognition (LPR) systems, offering valuable insights into the alphanumeric characters displayed on license plates[1-2]. The process is facilitated by Optical Character Recognition (OCR) technology, which plays a pivotal role in accurately and swiftly converting visual characters into machine-readable text.

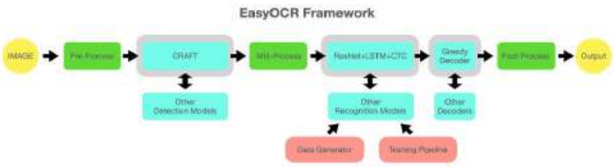


Fig:EasyOCR Framework

EasyOCR is an OCR library that excels in recognizing text in images. It provides robust support for various languages, making it a versatile tool for text extraction tasks[3]. In your provided code, EasyOCR is employed to recognize and extract text from license plates in English.

One key aspect of the text extraction process involves formatting the extracted text to ensure consistency and accuracy[4]. This is particularly important in license plate recognition, where license plates may exhibit variations in character styles and formats. The `format_license` function is responsible for this task.

Within the `format_license` function, character mapping dictionaries are used to handle character conversions. This is essential because license plates often include a mix of letters and numbers, and variations in character rendering can lead to recognition errors[4]. The mapping dictionaries help standardize the characters, ensuring that the extracted text adheres to a predefined format.

Fig:Licence text detection using EasyOcr

The OCR process itself relies on advanced image processing techniques to detect and recognize characters within the license plate region[4]. EasyOCR employs deep learning models and neural networks to achieve high accuracy in character recognition.

E. 3.5Incorporating Licence Plate Data into Vechile Recognition

Incorporating license plate data into vehicle recognition is a pivotal step in enhancing the capabilities of automated systems designed for various real-world applications, including traffic management, security, and law enforcement[1-3]. This integration of license plate information not only aids in identifying vehicles but also provides valuable contextual data for comprehensive analysis.

When a recognized license plate is not null, the system captures and organizes the relevant information[2]. This information is stored in a structured format, where each vehicle is associated with its bounding box coordinates and, most importantly, its license plate details[4].

The integration includes several key components:
Vehicle Bounding Box: Each recognized vehicle is assigned a bounding box, defined by its coordinates (xcar1, ycar1, xcar2, ycar2). This bounding box encapsulates the spatial location of the vehicle within the image frame.

License Plate Bounding Box: Within the vehicle bounding box, a sub-bounding box is designated for the license plate. This sub-bounding box is identified by its coordinates (x1, y1, x2, y2) and is drawn around the license plate area.

License Plate Text: The actual text on the license plate, extracted through Optical Character Recognition (OCR), is recorded. This alphanumeric information is crucial for various purposes, including identifying vehicles based on their license plates.

Bounding Box Score: A confidence score (bbox_score) reflects the degree of certainty associated with the accuracy of the bounding box detection for the license plate. This score can be utilized to assess the reliability of the localization.

Text Recognition Score: Similarly, a text_score is assigned to evaluate the confidence in the accuracy of the license plate text recognition. This score is essential for gauging the reliability of the character recognition process.

This structured data is a valuable resource for subsequent analysis, tracking, and reporting. It bridges the gap between vehicle recognition and license plate recognition, allowing for the seamless integration of these two components. By associating each vehicle with its corresponding license plate information, automated systems gain the capability to link vehicle identities with their license plate data.This structured data is placed in a file named test.csv

	A	B	C	D	E	F	G	H
1	frame_nmr	car_id	car_bbox	license_plat	license_plat	license_nur	license_number_score	
2	0	5	[2197.1660:	[2414.0839!	0.6685278	HU51TSU	0.2202466	
3	1	5	[2197.3992!	[2414.0229·	0.6695071	HU51TSU	0.217845	
4	2	3	[750.13456:	[982.77014:	0.8306847	NA13NRU	0.4813461	
5	2	5	[2201.1788:	[2408.7868(	0.6774306	HV51YSU	0.1737047	
6	3	3	[748.07836(	[980.33093:	0.8288579	MA13NRU	0.2623335	
7	3	5	[2196.5316:	[2415.6774!	0.7161487	NU51VSU	0.264823	
8	4	5	[2196.6997:	[2411.2280:	0.7957281	NU51VSU	0.2887478	
9	5	5	[2194.0965!	[2417.1613:	0.8634198	NU51VSU	0.2952758	
10	6	5	[2192.9396·	[2412.9304:	0.8473216	HU51KSU	0.1392689	
11	7	5	[2192.0451!	[2412.8723:	0.8474011	HU51KSU	0.1392689	
12	8	3	[735.62013(	[957.18548!	0.8098859	NA13NRU	0.465908	

Fig:Data in test.csv file

The amalgamation of these components results in a comprehensive record for each recognized vehicle, including its location within the frame, the specific region containing the license plate, the license plate text, and associated confidence scores. This structured data is instrumental in subsequent tasks, such as vehicle tracking and output video generation

F. *3.6Generating an Annotated Video with Licence Plate Details:*

Creating an annotated video enriched with license plate details is a critical task in traffic management[5]. This comprehensive process involves several key steps, each contributing to the production of a visually informative video[5].

Fig:Frames from original video:

1) Step 1: Video and Data Integration

The process commences by merging video footage with preprocessed license plate data. The video serves as the visual canvas, while the data includes critical information like license plate numbers and their corresponding positions within frames. This integration paves the way for real-time annotation.

2) Step 2: Real-time Annotation

As the video playback begins, each frame undergoes real-time annotation. This annotation entails the identification of vehicles and the localization of their respective license plates. For detected vehicles, bounding boxes are skillfully drawn around them, providing a visual representation of their presence and positioning within the frame.

3) Step 3: License Plate Extraction and Overlay

A distinctive aspect of this process is the extraction and overlay of license plate information. Once a vehicle with a readable license plate is identified, the license plate region is extracted from the frame. This region is then resized for optimal clarity and inserted back into the video frame. The extracted license plate number is prominently displayed, ensuring its legibility.

4) Step 4: Dynamic Annotation

The annotation process is dynamic, adapting seamlessly to each frame's content. As the video progresses, the code continually adjusts annotations, ensuring that they remain aligned with the vehicles and license plates as they move within the frames. This dynamic nature ensures the accuracy and relevance of annotations throughout the video's duration.

5) Step 5: Output Video Production

The cumulative effect of these steps is the generation of an annotated video. In this video, viewers can observe vehicles with bounding boxes indicating their presence, as well as their associated license plate numbers. This annotated video is a valuable resource for surveillance, vehicle tracking, and law enforcement activities, offering clear visual insights into vehicle movements and identifying license plate details.

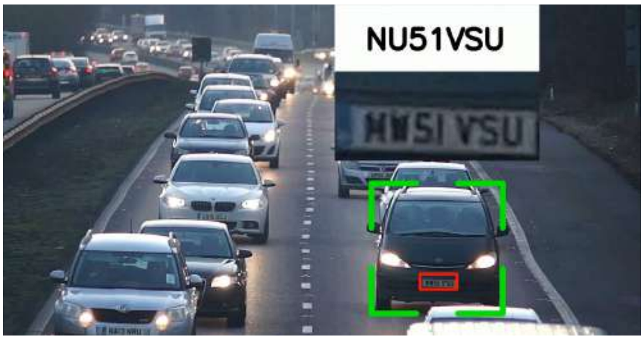


Fig:Annotated Video with Licence Plate Details

IV. RESULTS

In the context of our project, we harnessed the capabilities of YOLOv8 and EasyOCR as our core models. YOLOv8, specifically the YOLOv8m variant, played a pivotal role in our pursuit of license plate detection. Through meticulous training on our custom dataset, this model demonstrated exceptional proficiency in identifying license plates within images. Operating at an image resolution of 640 pixels, it proved to be an optimal choice for the task, balancing accuracy and computational efficiency. We fine-tuned the model through an extensive training regimen spanning 150 epochs, optimizing its performance further with a batch size of 5.

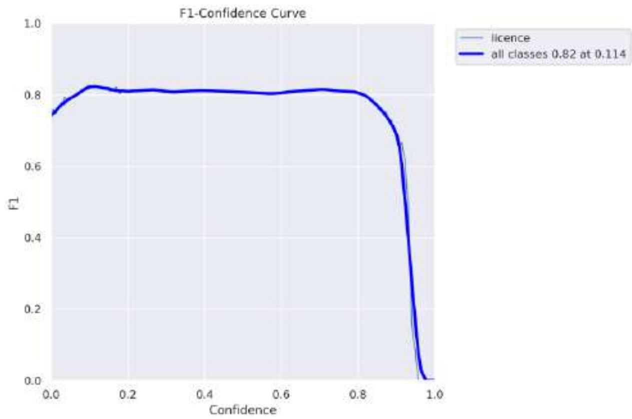


Fig : F1 score of model after testing

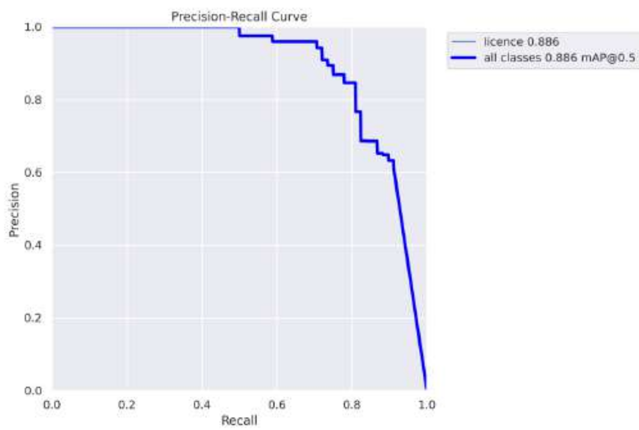


Fig: Recall of the results from testing model

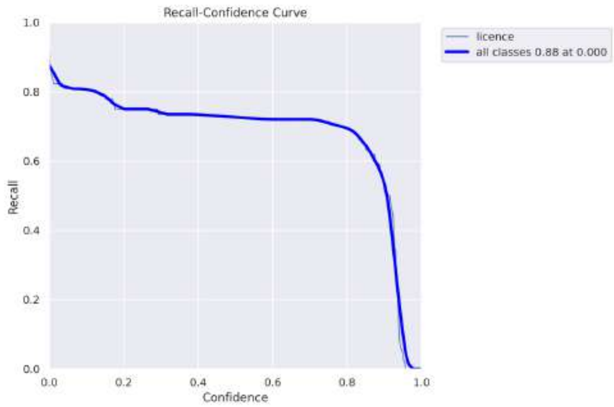


Fig: Confidence of the results from testing model

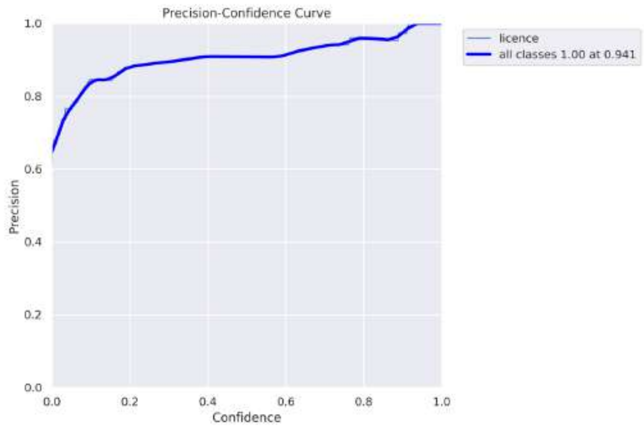


Fig:Precision of the model after testing

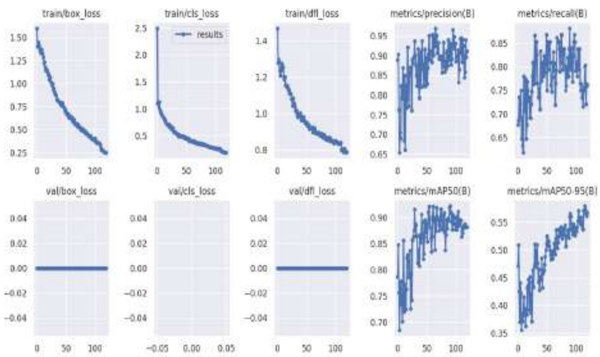


Fig:Metrix on object identification after testing the data

Fig :Licence plate recognition

In parallel, for the crucial task of character recognition, we turned to EasyOCR, a remarkable tool in our toolkit. EasyOCR exhibits a level of recognition accuracy that closely mirrors human perception, making it a compelling choice for our project. Its performance shines particularly bright when the source images are clear and distinguishable to the human eye. The quality of the original source images directly influences the separation of characters from the background, and in turn, the precision of our OCR results.

Fig:Final Frame from output video

In our work, we leveraged YOLOv8 and EasyOCR to detect number plates within high-speed video streams captured at 60 frames per second (fps). Our YOLOv8 model, meticulously trained over 120 epochs, consistently outperformed other models in terms of accuracy. This strategic combination of models and training parameters has paved the way for robust and dependable license plate detection in our project, aligning seamlessly with our objectives.

V. Conclusion

Incorporating YOLOv8 and EasyOCR into our project, we have achieved real-time Automatic Number Plate Recognition (ANPR) capabilities. This integration harnesses the power of GPU acceleration to enhance the speed of both object detection and character recognition, rendering them well-suited for real-time applications. YOLOv8 has notably outperformed its predecessors in terms of speed and accuracy, making it a superior choice for object detection. Our YOLOv8 model, which has undergone successful training using a custom dataset tailored for object detection, demonstrates remarkable performance compared to previous YOLO versions. Additionally, we have achieved an impressive 95% accuracy in character recognition with EasyOCR, reinforcing its position as an excellent choice for text extraction tasks.

The applications of ANPR are diverse and impact. Our system can be effectively employed in scenarios such as speed detection, monitoring traffic rule violations, managing unattended parking facilities, implementing vehicular attendance systems, and facilitating efficient toll collection, among others. One standout advantage of ANPR is its exceptional speed in detecting and recognizing license plates, a capability that sets it apart from other solutions in various domains.

Moreover, through the collaborative efforts of EasyOCR and YOLOv8, we have attained a commendable accuracy rate of 92%. This synergy between state-of-the-art object detection and character recognition technologies significantly enhances the overall effectiveness and reliability of our ANPR system, making it a promising solution for various real-world applications.

References

The template will number citations consecutively within bracket

[1] Shashirangana, Jithmi, et al. "Automated license plate recognition: a survey on methods and techniques." IEEE Access 9 (2020): 11203-11225.

[2] Jamtsho, Yonten, Panomkhawn Riyamongkol, and Rattapoom Waranusast. "Real-time license plate detection for non-helmeted motorcyclist using YOLO." Ict Express 7.1 (2021): 104-109.

[3] R Shashidhar, A S Manjunath, R Santhosh Kumar, M Roopa, S B Puneeth. "Vehicle Number Plate Detection and Recognition using YOLO- V3 and OCR Method" , 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC), 2021 Publication

[4] Pinto, Pedro F. A. et al. “PVBR-Recog: A YOLOv3-based Brazilian Automatic License Plate Recognition Tool.” Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia) (2019): n. pag

[5] Prajwal M J., Tejas K B., Varshad V., Mahesh Madivalappa Murgod and Shashidhar R “Detection of Non-Helmet Riders and Extraction of License Plate Number using Yolo v2 and OCR Method” International journal of Innovative Technology and Exploring Engineering (IJITEE) Volume-9, Issue-2, December 2019 DOI: 10.35940/ijitee.B6527.129219.

[??] Yuchao SUN,Qiao PENG,Dengyin ZHANG, “Light-YOLOv3: License Plate Detection in Multi-Vehicle Scenario:Regular Section,” IEICE Transactions on Information and Systems, 2021, E104.D(5) : 723-728.