International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Smart Traffic Surveillance: Helmet Detection and License Plate Recognition Using Deep Learning

Patlolla Sathvika Reddy
Computer Science Engineering
Institute of Aeronautical Engineering
Hyderabad, India
21951a05j1@iare.ac.in
0009-0002-7883-3961

Allam Shriya
Computer Science Engineering
Institute of Aeronautical Engineering
Hyderabad, India
21951a05k2@iare.ac.in
0009-0003-8840-4607

Pinnamwar Sai Siddanth
Computer Science Engineering
Institute of Aeronautical Engineering
Hyderabad, India
21951a05g7@iare.ac.in
0009-0006-4600

Dr. D Durga Bhavani
Computer Science Engineering
Institute of Aeronautical Engineering
Hyderabad, India
d.durgabhavani@iare.ac.in

Ms. D Rajani
Computer Science Engineering
Institute of Aeronautical Engineering
Hyderabad, India
dhaipulea.rajani@gmail.com
0000-0001-7001-5150

Abstract— Motorcyclists are particularly vulnerable to road accidents, often resulting in severe injuries or fatalities. Helmets are a proven safety measure that significantly reduces the risk of head injuries; however, many riders fail to comply with helmet laws, making enforcement a challenge for traffic authorities. This paper presents an advanced, automated traffic violation detection system that integrates real-time helmet detection and automatic license plate recognition to enhance road safety and streamline enforcement. Our system employs an optimized YOLOv3 model to efficiently detect motorcycles and determine whether riders are wearing helmets. Unlike conventional implementations, we enhance detection accuracy by fine-hyperparameter tuning YOLOv3 on a diverse dataset that includes various helmet types, different lighting conditions, and occlusions. To address challenges such as low-light environments, we incorporate preprocessing techniques, including contrast enhancement and adaptive thresholding, improving detection performance under suboptimal conditions. For license plate recognition, we utilize EasyOCR, further improved through custom preprocessing steps such as noise reduction and edge enhancement and thresholding enabling better recognition of partially occluded or low-quality license plates. Upon detecting a violation, the system automatically extracts and logs the license plate details into a structured database, facilitating streamlined enforcement and legal action. Our experimental results demonstrate increased accuracy while reducing inference time compared to existing methods, making this system a scalable and deployable solution for real-time traffic monitoring. By automating violation detection and reporting, the proposed approach reduces the burden on law enforcement while encouraging greater compliance with helmet laws, ultimately contributing to safer roads.

Keywords— YOLO, EasyOCR, preprocessing techniques, traffic regulations and real-time monitoring.

I. INTRODUCTION

Motorcycle accidents represent a significant global public health issue, contributing greatly to road traffic injuries and fatalities. The World Health Organization (WHO) reports that motorcyclists are 27 times more likely to die in accidents per mile traveled compared to car occupants (scroll.in,2024). Wearing helmets is a highly effective measure for reducing the severity of injuries, lowering the risk of death by 37% and head injuries by 69%. However, compliance with helmet laws varies significantly, especially in regions with high motorcycle use and weak law enforcement (timesofindia,2024). Enforcing helmet laws poses a challenge due to the difficulty of monitoring non-compliant riders in real-time. Traditional methods, such as police manually checking for helmet use, are labor-intensive and impractical in busy or congested traffic conditions. These reactive approaches rely on direct observation, limiting their overall effectiveness in promoting widespread helmet use. Research by (Chiverton,2012) underscores the importance of detecting helmet use and identifying motorcycles to enhance rider safety. As motorcycle accidents continue to rise, there is a growing need for systems capable of detecting helmet use. Recent advancements in machine learning, particularly within the realm of artificial intelligence, have led to the development of sophisticated safety systems, including real-time helmet detection and number plate of vehicles recognition using deep learning techniques.

A system of this kind employs the YOLOv3 algorithm, which is exceptionally effective for detecting objects in real time to ascertain whether a motorcycle rider is wearing a helmet and to recognize license plates. The speed and accuracy of the YOLOv3 algorithm make it ideal for applications that require quick detection. Furthermore, Optical Character Recognition (OCR) technology is employed to read license plate numbers, especially when riders are not wearing helmets. Deep learning technologies are key in enhancing the enforcement of safety regulations, and Dynamic helmet monitoring and license plate identification are excellent examples of their application. By integrating these methods, a reliable, real-time system can be developed to improve traffic law enforcement and potentially reduce road-related fatalities and injuries. The system continuously analyzes visual data, such as the presence of helmets and the visibility of license plates, to ensure compliance with safety regulations. Deep learning algorithms play a critical role in improving detection accuracy, delivering precise results in both helmet monitoring and license plate identification. The system aims not only to detect helmet violations but also to capture and record the license plate information of offenders, thereby creating a more secure and efficient enforcement mechanism.

II. RELATED WORK

These techniques help identify geometric shapes, analyze local object features, and capture texture details in images. (Chiverton,2012) introduced the first automated system to detect motorcyclists without helmets using an SVM classifier. While pioneering, this system faced challenges, such as high misclassification rates and inadequate motorcyclist identification, achieving a accuracy detection of 92.43% on real-world surveillance footage.

To address these limitations, (Silva et al.,2013) refined the approach by initially identifying motorcyclists using an SVM classifier with LBP features, followed by helmet detection using descriptors that combine the Hough transform, HOG, and LBP. Test was conducted on a dataset of 255 images which was taken from public roads, the system achieved a 91.37% accuracy rate for helmet detection. (Silva et al.,2014) extended this method using a multilayer perceptron (MLP) classifier and HOG descriptors, improving accuracy to 93.45%. Concurrently, (Waranusast et al.,2013) implemented a KNN classifier that utilized region properties for classification. Despite difficulties with occlusions, this method involved counting and segmenting motorcyclists' heads using projection profiling and helmet detection using KNN classifier based on features extracted from segmented head regions. The results indicated an average detection accuracy of 84% for the near lane, 68% for the far lane, and 74% for both lanes combined.

Additional studies (Bhaskar et al., 2014) explored various image processing techniques, such as HOG features and classifiers, for helmet detection. (Marayatr et al.,2015) developed a vehicle detection and tracking method that (Mukhtar et al.,2015) adapted for helmet detection using HOG features. They provided one of the early foundational approaches to helmet detection using image processing techniques. (Dalal et al.,2005) significantly impacted subsequent helmet detection research with their introduction of the (HOG) method for person detection. (Rubaiyat et al.,2016) expanded on this by developing an automatic helmet detection system for construction safety, demonstrating the adaptability of image processing techniques.

(Doungmala et al.,2016) introduced a method employing a decision tree classifier with AdaBoost for detecting half and full helmets, although this was limited to sparse traffic videos and did not include initial motorcycle detection. (Selvathi et al.,2017) contributed with an intelligent transportation system aimed at preventing and detecting accidents, while (Tapadar et al.,2018) integrated accident and alcohol detection into smart helmets, showcasing a comprehensive approach to motorcyclist safety. Recent advancements in deep learning have further enhanced helmet detection systems.

(Rohith et al.,2019) developed a system utilizing the Caffe model to detect motorcycles and riders, achieving 76% accuracy through an overlap check. For helmet detection, the InceptionV3 model was used, reaching 81% accuracy with modified pre-trained weights. (Siebert et al.,2020) applied a deep learning approach using Fast-R-CNN, achieving notable improvements over traditional methods. (Liu et al.,2022) developed a deep learning network incorporating a Residual Transformer-Spatial Attention mechanism, improving the system's accuracy in detecting helmet-wearing motorcyclists. Further innovations (Tran et al.,2023) have focused on combining custom tracking frameworks, such as YOLO, with deep learning techniques like CNNs to better address helmet rule violations. These efforts demonstrate ongoing advancements aimed at enhancing road safety through automated helmet detection systems.

III. PROPOSED SYSTEM

The proposed system employs the YOLO (You Only Look Once) algorithm (Tran et al,2023), a real-time object detection method that efficiently identifies multiple objects in a single pass through a convolutional neural network. To enhance accuracy in challenging conditions like low-light and occlusions, the system integrates custom preprocessing techniques. YOLO is selected for its optimal balance between speed and accuracy, enabling the detection of the rider, license plate, and rider's head. A Darknet53 classifier then determines whether the rider is wearing a helmet. If a violation is detected, the system uses Optical Character Recognition (OCR) to extract the license plate number and saves it for enforcement. Additionally, an image of the rider is captured and stored for further processing.

Fig 3.1 System Architecture

A. *Methodology*

The proposed system is structured around the following key components:

1)Data Collection and Preprocessing: Images of motorcycle riders, both with and without helmets, and vehicles with visible license plates will be gathered from multiple sources such as traffic cameras, public datasets, and manually captured photographs. These images will be preprocessed using tools like OpenCV(Tran et al.,2023) which undergoes preprocessing techniques such as contrast enhancement, histogram equalization, and noise reduction to improve quality and detection under low-light and occluded conditions, standardize dimensions, and label them appropriately (helmet, no-helmet, license-plate). Annotation tools such as LabelImg will assist in the labeling process.

2) Model Selection – YOLO: YOLOv3, from the YOLO (You Only Look Once) series, will be employed due to its high-speed processing and accuracy in object detection tasks. This makes it ideal for real-time applications. A pre-trained YOLO model will be fine-tuned with the gathered dataset to specialize in detecting helmets and license plates. YOLOv3 utilizes a backbone called Darknet-53, which includes 53 convolutional layers for feature extraction. To enhance performance, it incorporates Batch Normalization, which improves training stability and speed, and Leaky ReLU activations to introduce non-linearity without causing inactive neurons. The model architecture is organized into a neck and a head. The neck uses Feature Pyramid Networks (FPN) to merge features from different scales, enabling better detection of objects of varying sizes. This multi-scale feature fusion is essential for boosting detection accuracy, especially for smaller objects. The head of YOLOv3 is designed to predict bounding boxes and classify objects at three distinct scales, improving its accuracy and adaptability for detecting objects of different sizes.

3) Training Process: To increase the model's ability to handle different input variations, *data augmentation* techniques which incudes rotation, flipping, scaling, and brightness adjustments will be applied to the training dataset. These methods enhance the robustness of the model.

Transfer Learning: The training process will begin with a pre-trained YOLOv3 model on related datasets to leverage the existing learned features, facilitating quicker and more effective learning.

Training Configuration*:* The model will be trained using an appropriate batch size, number of epochs, learning rate, and optimizer. It will aim to minimize loss functions including objectness loss (indicating whether an object is present or not), classification loss (related to the accuracy of class labels), and bounding box regression loss (pertaining to the precision of bounding box predictions).

4)Detection Pipeline (Inference): The system will process inputs from various sources or static images, utilizing the trained YOLOv3 model for object detection. It will identify individuals on vehicles, helmets, and license plates. The inference function preprocesses the input image, runs the model to obtain predictions, and applies post-processing to refine these predictions.

Fig3.2: An illustration of images captured by a camera, depicting a rider with and without a helmet.

Human Detection with Vehicle: The selected frame is input into the YOLOv2 object detection model, configured to detect "Motorbike" and "Person" classes. The output will include the detected objects with bounding boxes, probability values, and detection confidence. The identified person is automatically cropped from the bounding box for further analysis. With the Image AI library, identified items or objects are isolated and stored as separate images, named consecutively with their category and image identifier., such as "motorcycle-1," "motorcycle-2," or "person-1," "person-2." These images are then stored in a dictionary for subsequent processing.

Helmet Detection: To reduce false positives during testing, the image is cropped to include only the upper one-fourth portion, eliminating errors when the rider is holding the helmet or placing it on the motorcycle. The helmet detection utilizes the darknet53 image classifier, pre-trained on the ImageNet dataset with 3054 helmeted/non-helmeted images, in conjunction with the YOLOv3 model with adaptive thresholding to differentiate helmets from background noise in complex environments. (to detect the head region) to determine whether the rider is wearing a helmet.

License Plate Detection: When the helmet detection model identifies that the rider is not wearing a helmet, the license plate recognition module is activated. The YOLO model, which has been specifically trained for detecting license plates, analyzes the image and provides the coordinates of the license plate for subsequent actions. If the helmet is detected, the image processing is halted.

Result Layer: The final result comprises images with annotations, displaying bounding boxes around detected helmets and license plates, accompanied by the respective class labels (helmet, no-helmet, and license plate).

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

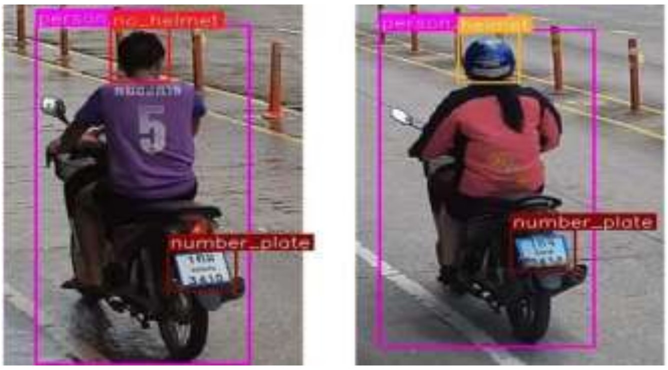

Fig3.3: It demonstrates that if a person is not wearing a helmet, the bike's license plate is extracted

5)License Plate Recognition (LPR): After detecting a license plate, the corresponding region is extracted and processed by an Optical Character Recognition (OCR) system. Tesseract OCR, an open-source OCR engine, can be employed to identify the characters on the plate and output the license number. To enhance OCR accuracy, noise reduction techniques such as Gaussian filtering, along with edge enhancement and Adaptive Thresholding are applied to improve text clarity, especially in blurred or low-resolution license plates.

6)Alert and Notification System: In cases where a rider is identified as not wearing a helmet or a specific license plate needs to be flagged (e.g., for law enforcement), the system can initiate alerts or notifications. These alerts can be displayed on a screen, sent via SMS, or logged for future reference.

7)Data Storage and Analytics: Once a helmet violation is detected and the license plate is successfully recognized, the system stores the rider's image along with the license plate information in a designated folder. Analytical tools can then be used to generate reports on helmet compliance trends, which can support road safety initiatives and policy decisions.

B. Algorithm

• First, the raw images are preprocessed by resizing them to meet specific criteria. After preprocessing (using OpenCV), the training model is utilized to predict object classes.
• In the YOLOv3 algorithm, object detection is approached as a regression task. The image or frame is split into an S x S grid, with each cell being accountable for identifying an object if its center lies within that cell. Every grid cell produces a bounding box, a confidence score, and a class probability map, as illustrated in Figure 3.4

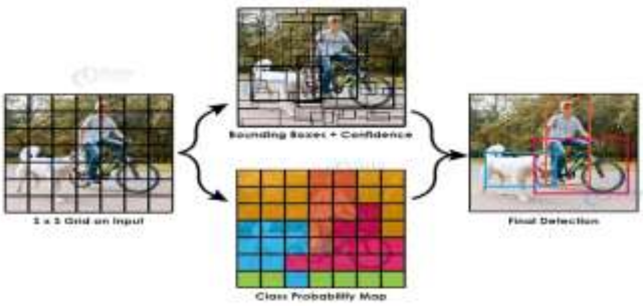

Fig 3.4: Object Detection with class annotations

• For cells that contain the center of an object, five bounding box parameters are predicted: (cx, cy, bw, bh, c). Here, (cx, cy) represent the coordinates of the object's center relative to the grid cell, while (bw, bh) specify the object's width and height relative to the image dimensions. The confidence score, denoted by c, reflects the accuracy of the prediction and is measured by the IoU (Intersection over Union) between the predicted and actual boxes, ranging from 0 to 1. The depiction of the algorithm is presented in Figure 3.5.
• The class probability indicates the likelihood of the object belonging to a specific category. YOLOv3 uses a two-class model to distinguish between a person with a helmet, a person without a helmet, and a license plate.
• The confidence score for each predicted bounding box is calculated. This score is a combination of the objectness score (probability that an object is present) and the class probability.
• Max(Class Probabilities): The class with the highest probability among those predicted for the bounding box is selected, representing the most likely class of the detected object.

<u>Confidence Score=Objectness Score× max(Class Probabilities)</u>

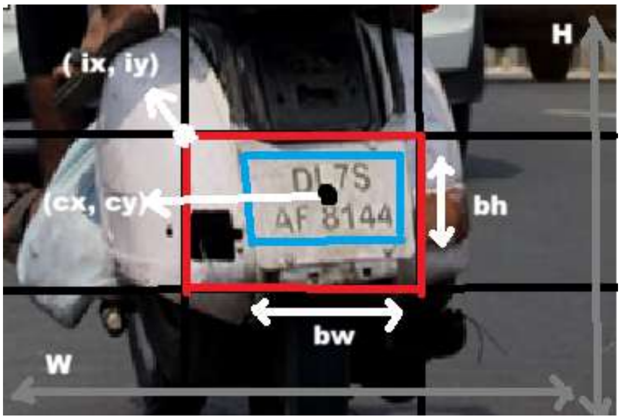

Fig3.5: An example using 3 × 3 grid cells is provided to illustrate the YOLO algorithm. (For a clearer understanding of the color references in this figure, please read the above description)

• In Fig3.5, The red rectangular cell of the grid is tasked with predicting the license plate since it encompasses the center of the License plate, marked by a circle filled with black colour. The coordinates (ix, iy) specify the initial position of the grid cell containing the LP's center (cx, cy).
• Prior to training the model, the datasets must be manually annotated with bounding box information for three categories: Rider, Helmet, and Plate. This annotation process was carried out using LabelImg software, as YOLO requires processed data. The software produces the starting coordinates (s0, e0) and ending coordinates (s1, e1), which were then normalized between 0 and 1 to conform to the YOLO format (s, e, w, h) using Eqs. (1)-(4):

$$s = \frac{(s0 + s1)}{2 * W} \quad (1)$$

$$e = \frac{(e0 + e1)}{2 * H} \quad (2)$$

$$w = \frac{(s1 - s0)}{W} \quad (3)$$

$$h = \frac{(e1 - e0)}{H} \quad (4)$$

• During prediction, the YOLO model outputs the center coordinates (cS , cE) along with the width (w) and height (h) of the bounding box. However, the proposed algorithm for automatically detecting license plates (LP) of motorcyclists without helmets requires the starting coordinates (s0, e0), which are obtained using Eqs. (5)–(6). The ending coordinates (s1, e1) are then determined using Eqs. (7)-(8):

$$s0 = cS - (w/2) \quad (5)$$

$$e0 = cE - (h/2) \quad (6)$$

$$s1 = s0 + w \quad (7)$$

$$e1 = e0 + h \quad (8)$$

• *Thresholding:* To minimize false positives, predictions that have confidence scores falling below a specific threshold are disregarded. Check if the confidence score meets a predefined threshold (50%).

- **Yes**: If the confidence score is above the threshold, proceed to the next step.
- **No**: If the confidence score is below the threshold, discard the bounding box (give up bounding box).

• *Post -processing (Non-Maximum Suppression (NMS)):* NMS helps in removing overlapping bounding boxes with lower confidence levels, keeping only the most relevant ones. This method calculates the Intersection over Union (IoU) between bounding boxes to measure their overlap, which is essential for selecting the best candidates during NMS.

• *Output the target bounding box:* Once NMS is applied, the remaining bounding boxes are provided with their associated class labels and confidence levels.

• *Annotating the target type and confidence score:* The detected boxes are marked on the image with the corresponding class labels and confidence values.

C. Overview of Implementation

Step 1: Collection of Data: Import necessary libraries. Create a Tkinter window for image upload. Define a function to upload the image. Store the uploaded image for further processing.

Step 2: Detect Motorbike & Person: Load the YOLOv2 model. Detect objects (two wheeler and Person) in the uploaded image. Extract and save the detected objects as separate images. Store the details of the extracted images in separate files.

Step 3: Detect Helmet: Load the YOLOv3 model for head detection. Load the ResNet50 model for helmet classification. Crop the top one-fourth section of the person image. Detect and classify whether the rider is wearing a helmet.

IV. RESULTS AND DISCUSSION

A. Results

The proposed system uses the darknet53 classifier for image processing and classification with the YOLO model to detect the rider without helmet and license plate. Then extract the license plate characters using OCR. This system is designed to enhance road safety by automatically detecting riders who are violating the rules such as travelling without helmet and capturing their license plate information for further action.

The system first takes the input as an image which we upload in the GUI of Tkinter window.

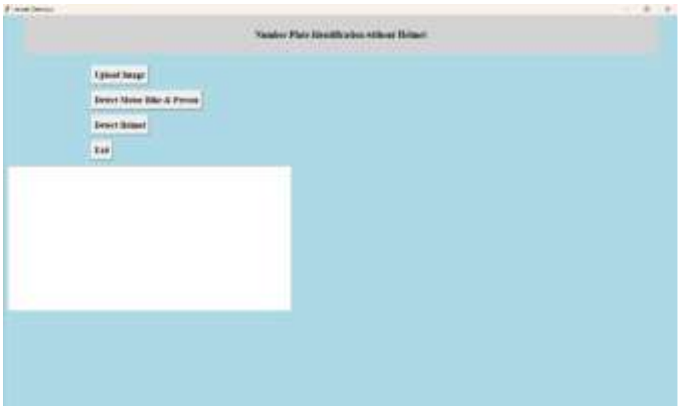

Fig 4.1 Execution Frame

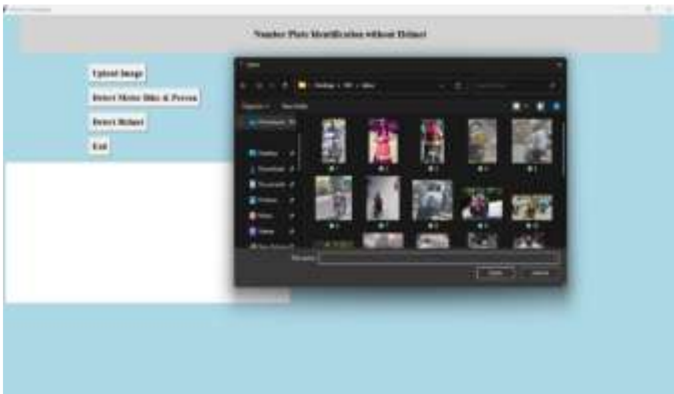

Fig 4.2 Upload Image

Once the image is uploaded click the "Detect Motorbike & rider" button to identify whether the image features a rider with a motorbike or not.

Fig 4.3 Detect Motorbike and Person

If the picture contains both a person and a bike, as detected by YOLO with its confidence score, proceed by clicking the "Detect Helmet" button to check if the rider is wearing a helmet.

Fig 4.4 Helmet Detection(Not there in frame)

If the application determines that the rider is not wearing a helmet, it will extract the number from the vehicle's license plate and display it in the adjacent text area for further processing.

Fig 4.5: Helmet detected

If the rider is wearing a helmet, the application will label the helmet around the person's head with the confidence score and stop further scanning, including the license plate.

B. Discussion

1)Training: The training process for the detection system was conducted using a Nvidia GTX 1070 graphics card, which took approximately six hours for each of the two categories involved. This training resulted in weight files around 250 MB each, which are subsequently loaded into the detection program.

The YOLOv3 training was divided into two parts: the first part focused on detecting bikes, persons, and the head region, while the second part targeted the motorcycle license plate. The graphs below (fig 4.6,4.7) depict the trend of the loss function. The model was trained over 4000 iterations, resulting in a final loss of 0.044 for the initial weight configuration and slightly below 0.158 for the subsequent one.

For helmet detection, the head region was trained to distinguish between helmets and non-helmets using a binary classifier. This was achieved by employing a ResNet50 model, which was pre-trained on ImageNet and then fine-tuned on the helmet dataset through transfer learning. During training, 20% of the dataset was reserved for validation. The training session was carried out in Jupyter with the help of the GTX 1070 graphics card. After training phase, the weights were saved and converted into the final code which is used for classification purposes.

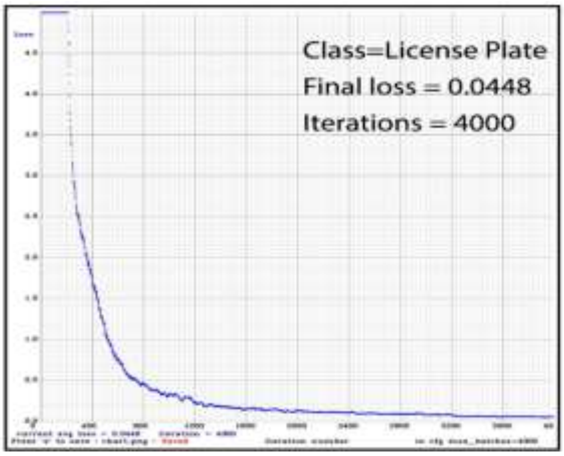

Fig.4.6 Class loss(license plate) vs Iterations

Fig.4.7 Class loss(head) vs Iterations

2)Tools and Platform Used: The project utilizes several tools and platforms to achieve real-time detection. YOLO versions is employed for object detection, specifically identifying the rider, license plate, and rider's head, while

ResNet50 is used for helmet classification. Tesseract OCR is integrated to extract plate numbers when the rider isn't wearing a helmet. Key libraries such as OpenCV, PyTorch, tkinter (GUI interface), keras with tensorflow (Backend to construct CNN) and Torchvision facilitate image processing, deep learning model implementation, and data management. The development is done on platforms like Visual Studio Code or Jupyter Notebook, supported by a GPU-enabled system (preferably with NVIDIA CUDA support) for efficient processing.

3)Datasets Used: The project relies on datasets containing images of riders with and without helmets, along with annotated vehicle images for training the models, and uses pre-trained weights to enhance the detection and classification tasks.

4)Performance Evaluation: After training, the evaluation of the model's performance on the test dataset using various metrics is as follows:
Precision and Recall: Evaluate/Measure the accuracy and completeness of the detections
In our experiment, the classifier for distinguishing motorcycles from non-motorcycles achieved an accuracy of 97.58%, with a precision and recall both at 98.56% on the test dataset. The classifier for detecting whether a person riding a vehicle was wearing a helmet or not attained an accuracy of 98.89%, a precision of 99.30%, and a recall of 98.41% on the test dataset. Consequently, the overall accuracy for identifying motorcyclists without helmets was calculated as the product of the two classifiers' accuracies, resulting in 96.17% (97.58% * 98.56%). For the optical character recognition (OCR) component, achieved an accuracy of 93.36% on the test images. Table I presents the performance of each classifier on the test data. The metrics were computed using the following formulas:

$$\text{Accuracy} = \frac{\text{(Number of correctly identified samples)}}{\text{(Total number of samples)}}$$

$$\text{Precision} = \frac{\text{(Number of positive samples identified positive)}}{\text{(Total number of samples identified as positive)}}$$

$$\text{Recall} = \frac{\text{(Number of positive samples identified as positive)}}{\text{(Total number of actual positive samples)}}$$

Classifier	Performance (%)					
	Accuracy		Precision		Recall	
	Yolov3	Yolo v3(opt)	Yolov3	Yolov3(opt)	Yolov3	Yolov3(opt)
Motorcycle vs nonmotorcycle Classifier	86.3	97.58	89.05	98.56	88.97	98.56
Helmet vs non helmet classifier in low light	81.2	98.89	81.04	99.30	82.43	98.41
Easy OCR (for occluded plates)	78.9	86.65	79.32	89.65	80.23	87.87

opt – Optimized through YOLOv3 Hyperparameter Tuning, Adaptive thresholding before EasyOCR

Table I: Performance of Each Classifier On Test Data

V. CONCLUSION

A Non-Helmet Rider Detection system for motorcycles provides a valuable means to enhance road safety and bolster traffic law enforcement. The system processes image as input and detects whether the motorcycle rider is wearing a helmet. If the rider is not, the system identifies and extracts the motorcycle's license plate number. This detection utilizes fine-tuned YOLOv3 and EasyOCR, optimized with custom preprocessing techniques to identify motorcycles, riders, and helmets. To ensure proper enforcement, the system also captures the number plate of the offending motorcycle using OCR system with thresholding for better text extraction. The extracted plate number, along with the corresponding frame, is recorded for further use. The project aims for high accuracy in detecting helmet violations and achieves this by improving both detection and identification processes. Future enhancements in system accuracy and performance, especially under challenging conditions such as low lighting or adverse weather, can be realized through advanced preprocessing techniques like contrast enhancement, image denoising, noise reduction, and edge detection. This will help ensure clearer images for detection.

A user-friendly web interface has been developed to ensure easy access and smooth navigation, allowing users to view detection results effortlessly. Additionally, the system addresses key non-functional requirements such as performance, reliability, security, and scalability, ensuring that the platform is not only effective but also robust and user-friendly. In conclusion, the successful deployment of this system could greatly enhance road safety, reduce traffic accidents, and support the enforcement of traffic regulations, contributing to the overall safety of road users.

VI. FUTURE SCOPE

The project can be integrated with traffic cameras to detect helmet use in real-time, enabling the Transport Office to access license plate data and enforce helmet laws by issuing fines. By incorporating automated license plate detection, the system could generate challans for helmet violations and could be expanded to detect other traffic offenses, such as seat belt non-compliance and mobile phone usage while driving. Additionally, the codebase could be converted into a GUI integrated with an online platform that live streams traffic monitoring by authorities, contributing to improved road safety. A future enhancement could include a system that improve scalability by integrating a multi-threaded processing approach for large-scale deployments.

REFERENCES

[1] J.Chiverton, “Helmet Presence Classification with Motorcycle Detection And Tracking”, IET Intelligent Transport Systems,Vol. 6, Issue 3, pp. 259–269, March 2012.

[2] Romuere Silva, Kelson Aires, Thiago Santos, Kalyf Abdala, Rodrigo Veras, Andr´e Soares, “Automatic Detection Of Motorcyclists without Helmet”, 2013 XXXIX Latin America Computing Conference (CLEI).IEEE,2013.

[3] Rattapoom Waranusast, Nannaphat Bundon, Vasan Timtong and Chainarong Tangnoi, “Machine Vision techniques for Motorcycle Safety Helmet Detection”, 28th International Conference on Image and Vision Computing New Zealand, pp 35-40, IVCNZ 2013.

[4] M. K. A. Mohd Rasli, N. K. Madzhi, and J. Johari, "Smart helmet with sensors for accident prevention," in 2013 International Conference on Electrical, Electronics and System Engineering (ICEESE), Kuala Lumpur, 2013.

[5] P. K. Bhaskar and S.-P. Yong, "Image processing-based vehicle detection and tracking method," in 2014 International Conference on Computer and Information Sciences (ICCOINS), 2014.

[6] Romuere Silva, “Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers”, 27th SIBGRAPI Conference on Graphics, Patterns and Images.IEEE, 2014.

[7] Thepnimit Marayatr, Pinit Kumhom, “Motorcyclist‟s Helmet Wearing Detection Using Image Processing”, Advanced Materials Research Vol 931- 932,pp. 588-592,May-2014.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1

[9] R. R. V. e. Silva, K. R. T. Aires and R. d. M. S. Veras, "Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers," 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images, Rio de Janeiro, 2014, pp. 141-148.

[10] Amir Mukhtar, Tong Boon Tang, “Vision Based Motorcycle Detection using HOG features”, IEEE International Conference on Signal and Image Processing Applications (ICSIPA).IEEE, 2015.

[11] Abu H. M. Rubaiyat, Tanjin T. Toma, Masoumeh Kalantari-Khandani, “Automatic Detection of Helmet Uses for Construction Safety”, IEEE/WIC/ACM International Conference on Web Intelligence Workshops(WIW).IEEE, 2016.

[12] P. Doungmala and K. Klubsuwan, "Helmet Wearing Detection in Thailand Using Haar Like Feature and Circle Hough Transform on Image Processing," 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, 2016, pp. 611-614.

[13] D. Selvathi, P. Pavithra, and T. Preethi presented a study on an "Intelligent transportation system for accident prevention and detection," in 2017 International Conference on Intelligent Computing and Control Systems (ICICCS)

[14] S. Tapadar, S. Ray, H. N. Saha, A. K. Saha, and R. Karlose, "Accident and alcohol detection in Bluetooth-enabled smart helmets for motorbikes," was featured at the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018.

[15]Rohith C, Shilpa ,Parvathi Sanil Nair, Sneha Alphonsa, “An Efficient Helmet Detection For MVD using deep learning” in 2019 IEEE Xplore, Proceedings of the Third International Conference on Trends in Electronics and Informatics(ICOEI)

[16] Siebert and Lin, published a preprint on arXiv titled "Detecting Motorcycle Helmet Use with Deep Learning," arXiv preprint, 2020.

[17] Liu., Chen, & Wang, explored "Helmet Wearing Detection of Motorcycle Drivers Using Deep Learning Network with Residual Transformer-Spatial Attention," Drones, vol. 6, no. 12, 2022, pp. 415-425.

[18] Tran, Q. H., Nguyen, H. S. P., & Nguyen, D. Q., "Helmet Rule Violation Detection for Motorcyclists Using a Custom Tracking Framework," CVPR Workshops, 2023.

[19][Online]. Available: https://scroll.in/article/757365/road-accidents-kill382-in-india-every-day-1682-times-more-than-terrorism

[20] [Online]. Available:
http://timesofindia.indiatimes.com/city/chennai/98- 6-of-bikers-who-died-didnt-wear-a-helmet/articleshow/4790