# HELMET DETECTION FOR MVD

**A PROJECT REPORT**

***Submitted by***


**BINCY BIJU(TKM21MCA-2013 )**


***to***

**The APJ Abdul Kalam Technological University**

*in partial fulfilment of requirements for the award of degree of*

**MASTER OF COMPUTER APPLICATIONS**


Thangal Kunju Musaliar College of Engineering
Kerala

**DEPARTMENT OF COMPUTER APPLICATIONS**


**MAY 2023**

# DECLARATION

I hereby declare that the project report on **" Helmet Detection for MVD",** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Dr Fousia M Shamsudeen.** This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or theUniversity and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. The report has not been previously formed the basis for theaward of any degree, diploma or similar title of any other University.

Kollam **BINCY BIJU**

18/05/23

**DEPARTMENT OF COMPUTER APPLICATIONS**
**TKM COLLEGE OF ENGINEERING**

# C E R T I F I C A T E

This is to certify that, the project report entitled **"Helmet Detection for MVD"** submitted by **BINCY BIJU ( TKM21MCA-2013)** to the APJ Abdul Kalam Technological Universityin partial fulfillment of the requirements for the award of the degree of Master of Computer Applications, is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor                    Head of the Department                    External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia M Shamsudeen**, Head of the Department, for providing me with best facilities.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my advisor and coordinator **Prof. Natheera Beevi M** Department of Computer Applications, for her expert guidance, co-operation, and immense encouragement.

I would like to thank my project guide **Dr. Fousia M Shamsudeen**, Department of Computer Applications, who motivated me throughout the work of my project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

**BINCY BIJU**

# ABSTRACT

Helmet detection plays a crucial role in ensuring the safety of motorcyclists. Helmet detection system using the YOLOv5 architecture and integrate it into a Django web application is proposed. The system aims to automatically detect the presence of helmets in images or real-time video streams. The YOLOv5 model trained on a dataset consisting of helmet and non-helmet images, utilizing transfer learning for improved performance. The trained model is serialized and integrated into the Django application, allowing for seamless inference and prediction of helmet detection. API endpoints are defined to receive image inputs and return the detection results. The system includes data preprocessing steps to handle incoming images and optimize them for the model's input requirements. The deployment and integration of the model into the Django framework enable real-time helmet detection and provide a user-friendly interface for users to interact with the system. Extensive testing and evaluation demonstrate the effectiveness and accuracy of the helmet detection system. The developed solution has the potential to contribute significantly to promoting helmet usage and enhancing road safety for motorcyclists.

# Contents

# List of figures

# List of Abbreviations

**RCNN** Region based-Convolutional Neural Network

**YOLO** You Only Look Once

**ALPR** Automated License Plate Readers

# CHAPTER 1
# INTRODUCTION

In India, the number of accidents occurring each day is increasing rapidly. The two-wheelers account 25 percent of road crash deaths because of ignoring safety measures like wearing helmets while driving. More than two drivers are travelling in a two-wheeler is also a major reason. So here, we propose a framework for real-time detection of traffic rule violators who ride bikes without using a helmet. The road CCTV footage is used to detect whether a rider is wearing a helmet or not, using Deep Learning and Image Processing technology. The algorithm used here is faster RCNN, because it increases the detection rate of motorcycles, compared to other deep networks such as CNN, fast R-CNN and YOLO. The violator's vehicle's registration number is recognized from the vehicle using open-ALPR and an alert is sent to the nearby police station.

Here we mainly categorized into four areas they are,

- Motorcycle Detection
- Head Detection
- Helmet Detection

Since bikes are reasonable and a day by day method of transport, there has been a quick increment in bike mishaps because of the way that the majority of the motorcyclists don't wear a helmet which makes it an ever present risk each day to go by a bike. Over the most recent few years alone most of the deaths in accidents are due to damage in the head resulting in trauma to the skull or mind. In light of this, wearing a helmet is obligatory according to traffic rules, violation of which pulls in heavy fines. Inspite, an enormous number of motorcyclists don't comply with the standard. The police officer attempted to control this issue physically, however it is insufficient for the real circumstance. The requirement for security measures is an unquestionable requirement to decrease the number of deaths in road accidents, and use of helmets is a significant factor regarding safety.

A study that was conducted by the United Nations in 2015 estimated that the chances of surviving an accident rose by 42 percent on wearing a helmet. Even though helmets are for the safety of the riders, most of them avoid it due to reasons like "it spoils my hairstyle", "it feels uncomfortable"," good helmets are costly" or "it obstructs my peripheral vision". These reasons are not comparable to losing a life. The existing system for checking whether a rider is wearing a helmet or not is a checkpoint by police or other personnel to manually check each rider. In this system, there is an impossibility of riders evading checkpoints.

Thus, the importance of automatic systems in traffic control has been increased in recent years. Presently, all major urban areas already deployed huge video reconnaissance systems to keep a vigil on a wide assortment of dangers. In this way utilizing such a already existing system will be a cost-efficient arrangement, however, these frameworks include an enormous number of people whose performance is not significant for long periods of time. Recent studies have shown that human surveillance proves ineffective, as the span of checking of thus the importance of automatic systems in traffic control has been increased in recent years. Presently, all major urban areas already deployed huge video reconnaissance systems to keep a vigil on a wide assortment of dangers.

In this way utilizing such a already existing system will be a cost-efficient arrangement, however, these frameworks include an enormous number of people whose performance is not significant for long periods of time. Recent studies have shown that human surveillance proves ineffective, as the span of checking of recordings expands, the blunders made by people likewise increases. We aim to improve the usage of a traffic stream framework, others are to lessen the expense of human work and abate the reasons for a mishap. The ideal solution is to develop an electronic detection system that can be automated to recognize this kind of problem without human cost.

## 1.1 PROBLEM STATEMENT

Create an intelligent helmet detection system capable of accurately identifying whether individuals in images or video streams are wearing helmets or not, in order to enhance safety enforcement and mitigate the risk of head injuries in various contexts such as workplaces, recreational activities, and transportation. The system should provide real-time detection, high accuracy, and scalability to handle diverse environments and lighting conditions.

## 1.2 OBJECTIVES

- To develop a robust and accurate helmet detection algorithm that can effectively identify helmets in images or video frames.
- To train the helmet detection model using a comprehensive dataset that includes a wide range of helmet types, various head orientations, and diverse lighting conditions to ensure generalization.

- Optimize the helmet detection algorithm for real-time performance to enable fast and efficient detection in live video streams
- Implement the helmet detection system in a user-friendly interface, allowing easy integration into existing surveillance systems or safety enforcement frameworks.
- Evaluate the performance of the helmet detection system through rigorous testing and benchmarking against ground truth annotations, ensuring high accuracy and minimizing false detections or missed detections.
- Continuously refine and improve the helmet detection system based on user feedback and emerging advancements in computer vision techniques to enhance its overall effectiveness and reliability.
- Promote the adoption of the helmet detection system by raising awareness among relevant stakeholders and emphasizing the importance of helmet usage for safety purposes.

# CHAPTER 2

# LITERATURE SURVEY

Literature review is the comprehensive study and interpretation of literature that relates to a particular topic. When one uses literature review research questions are identified, then one seek to answer this research questions by searching for and analyzing relevant literature. Some importance of literature reviews is that new insights can be developed by the reanalyzing the results of the study. A literature review is both a summary and explanation of the complete and current state of knowledge on a topic as found in academic books and journal articles.

## 2.1 PURPOSE OF THE LITERATURE REVIEW

1. It gives readers easy access to research on a particular topic by selecting high quality articles or studies that are relevant, meaningful, important and valid and summarizing them into one complete report.
2. It provides an excellent starting point for researchers beginning to do research in a new area by forcing them to summarize, evaluate, and compare original research in that specific area.
3. It ensures that researchers do not duplicate work that has already been done.
4. It can provide clues as to where future research is heading or recommend areas on which to focus.
5. It highlights the key findings.
6. It identifies inconsistencies, gaps and contradictions in the literature.
7. It provides a constructive analysis of methodologies and approaches of researchers.

## 2.2. RELATED WORKS

Pathasu Doungmala, Katanyoo Klubsuwan [1] proposed a new helmet detection technique. The technique combines two methods for helmet detection to achieve better detection rate. The two methods are face detection using Haar like feature for detection between without helmet and full helmet and circle Hough transform for detection between without helmet and half helmet. The proposed full and half helmet detection and segmentation method used the visual face/nose/mouth/ left eye/right eye detection using Haar like feature and circle Hough transform identify classes of full and half helmet. This system overcame various issues raised by the complexity of full and half helmet detection problems. This system overcame various issues raised by the complexity of full and half helmet detection problems.

Madhuchhanda Dasgupta, Oishila Bandyopadhyay, Sanjay Chatterji [2] proposes a frame work for detection of single or multiple riders travel on a motor cycle without wearing helmet in the proposed approach at the first stage motor cycle riders are detected using YOLOv3 model. In the second stage a Convolutional Neural Network(CNN) based architecture has been proposed for helmet detection of motor cycle riders. This project aims to decrease the accidents caused due to not wearing the helmet it stores the image of the violated people along with a cropped number plate . It is then send through the mail to authorities. The algorithm used here is CNN, which is faster and accurate when compared to others.

Joseph Redmon, Ali Farhadi [3] introduce a YOLO9000, it is a state of the art, real time object detection system that can detect over 9000 object category. The improved model is YOLOv2, it is a state of the art on standard detection task like PASCAL VOC and COCO.YOLOv2 and YOLO9000 are the real time detection system. YOLOv2 is state of the art and faster than other detection system. It can be run at a variety of image sizes to provide a smooth tradeoff between speed and accuracy. YOLO9000 is a real time framework for detection more than 9000 object categories by jointly optimizing detection and classification.

Xiong, Yu and Cheng, Haozhi and Wang, Bo and Farhadi, Ali and Yuille, Alan [4] provides an extensive overview of deep learning techniques for object detection. It covers popular architectures like YOLO, Faster R-CNN, and SSD, explaining their underlying principles and methodologies. The paper discusses important aspects such as model architectures, training strategies, evaluation metrics, and benchmark datasets used in the object detection field. It also highlights recent advancements and trends in deep learning-based object detection, providing valuable insights for researchers and practitioners.

Islam, Md Zahangir and Bhuiyan, Md Zakirul Alam and Lu, Gang [5] focuses on recent advancements in deep learning-based object detection methods. It covers both single-stage and two-stage approaches, providing an overview of their architectures, loss functions, and techniques for improving detection accuracy and speed. The paper discusses various deep learning frameworks and their applications in object detection. Additionally, it examines datasets and evaluation metrics commonly used in the field, helping readers gain a comprehensive understanding of the state-of-the-art techniques in deep learning-based object detection.

Aditya Mandeep Vakani [6] has discussed "Automatic License Plate Recognition of Bikers with No Helmets" employs YOLO with pre-trained weights on the COCO dataset to first identify a motorcy cle, then a person, and then a check to see if the two overlap to identify the person as being on the i dentified motorcycle. Furthermore, the upper one-fourth region is divided into helmet and non-helm et categories using a 5-layer CNN structure based on YOLO-LITE. A functional prototype of a syst em that can identify bikers who disobey the law by not wearing helmets and put their own safety in danger has been created. A system like this can actively contribute to a decrease in the number of p eople who disregard traffic laws. YOLO's advantages include speed and real-time processing. The s ystem's limitation to operation in good illumination and clear weather is a downside.

Parasa Teja Sree [7] has discussed "Real Time Automatic Detection of Motorcyclists with and With out a Safety Helmet" which is a machine learning-based approach positioned to identify helmet usa ge among motorcyclists. Video frames attained from surveillance footage, the object detection-base d algorithm is trained to spot motorcycles and their helmet. Through various tools and methods corr esponding to Open CV and support vector classification, the desktop interface application is made p ossible to visualize the live streaming traffic surveillance footage which uses Support Vector Machi ne (SVM) and CNN. This system has an accuracy of 87%. The advantage is Greater accuracy in ge nerating image classification and recognition algorithms such as CNN will prove to be more benefi cial. The only subsidiary due to the usage of CNN is the requirement for larger data processing unit s which increases the corresponding time taken for training the model.

Nitin Nagori [8] and his team realized that the detection and classification of images and video requ ire features. And feature extraction manually is an invincible work and therefore used Convolutiona l Neural Network. CNN learns the whole image by extracting features using feature maps and has p roven to obtain better detection and classification. At the same time, this model needs a huge amou nt of data sets to train from the beginning. A small amount of data may underfit the data based on a given type of learning. So, they used transfer learning on the CNN model, Yolov3-tiny darknet trai ned in advance on the COCO (Common Objects in Context) dataset. With these features, high preci sion is acquired. The major drawback was that this model needed a lot of datasets.

Dikshant Manocha and team [9] they carried out their project in four steps. Stage 1: This step entail ed gathering data and classifying photos into positive (which includes those that need to be identifie d in real-time) and negative (which do not) (which are to be ignored in real-time). However, a lack of sample images restricts this step. Stage 2: Two situations must be educated into the machine. If t he rider is wearing a helmet, nothing needs to be done. If not, the system must use the HAAR file to look for the registration number plate and send it to optical character recognition if it is discovered ( OCR). Upon failure, a police officer on duty will receive an alert message with instructions on wha t to do. However, this limitation means that machine learning will take longer as the number of sam ple photos rises. However, system accuracy may suffer if the number of sample photos is lowered. Stage 3: At this point, the system has determined that the rider is not wearing a helmet. It has also i dentified the location of the license plate and called on OCR. By doing so, the real two-wheeler reg istration number for which a challan must be generated will be extracted. But this stage is constrain ed by inaccurate meteorological conditions. Stage 4: Web or mobile interfaces for paying fines wer e developed.

Pathasu Doungmala, Katanyoo Klubsuwan [10] proposed a new helmet detection technique is proposed in this paper. The technique combines two methods for helmet detection to achieve better detection rate. The two methods are face detection using Haar like feature for detection between without helmet and full helmet and circle Hough transform for detection between without helmet and half helmet. The proposed full and half helmet detection and segmentation method used the visual face/nose/mouth/ left eye/right eye detection using Haar like feature and circle Hough transform identify classes of full and half helmet. This system overcame various issues raised by the complexity of full andhalf helmet detection problems. This system overcame various issues raised by the complexityof full and half helmet detection problems.

Pathasu Doungmala and team [11] proposes a frame work for detection of single or multiple riders travel on a motor cycle without wearing helmet in the proposed approach at the first stage motor cycle riders are detected using YOLOv3 model. In the second stage a Convolutional Neural Network(CNN) based architecture has been proposed for helmet detection of motor cycle riders. This project aims to decrease the accidents caused due to not wearing the helmet it storesthe image of the violated people along with a cropped number plate . It is then send through the mail to authorities. The algorithm used here is CNN, which is faster and accurate when compared to others.

Wei Hou et al. [12] proposed a technique that used the moving object library and the data structure in the computer vision library to construct a vehicle video analysis system for the detection and tracking of cars on the road. CAMSHIFT algorithm was used to solve the problem of target deformation and partial occlusion. Gaussian Background model was used to obtain background. That experiments had verified that for vehicle detection, the threshold of 20 is the best value.

Apeksha P Kulkarni et al. [13] proposed the technique that efficiently managed to distinguish the vehicles from the surrounding environmental variability and improve the low-resolution videos through the Histogram equalization technique to maintain uniformity of videos in terms of resolution and also in removal of noise from videos. Accuracy gained for detected vehicles by that approach is 97.39% and for a vehicle, tracking is 98.26%.

Neelam Dwivedi et al. [14] was proposed he efficient approach for Unattended Object Detection by Contour Formation utilizing Background Subtraction. The proposed approach uses background subtraction for object detection with the ability to change the background frame if the detected object is found non- suspicious/harmless. Average Correct Object Detection Rate, average Object Success Rate, and average False Alarm Rate of the static and dynamic background are 70.83%, 67.25 %, and 35.41% respectively. It can be run at a variety of image sizes to provide a smooth tradeoff between speed and accuracy. YOLO9000 is a real time framework for detection more than 9000 objectcategories by jointly optimizing detection and classification.

# CHAPTER 3
# METHODOLOGY

Helmet detection using deep learning is a powerful and efficient approach that leverages advanced neural network architectures to automatically identify and locate helmets in images or videos. Deep learning algorithms, such as YOLOv5, have revolutionized the field of computer vision by enabling real-time and accurate object detection. In the context of helmet detection, this methodology plays a crucial role in ensuring safety compliance and enhancing situational awareness in various domains, including construction sites, sports, and industrial environments. By harnessing the capabilities of deep learning, helmet detection models can quickly and reliably identify instances of helmets, aiding in critical tasks such as safety monitoring, helmet usage analysis, and risk mitigation.

## 3.1 ALGORITHM

A helmet detection algorithm is a computer vision-based algorithm designed to identify and locate helmets in images or video frames. It is commonly used in various applications, such as workplace safety monitoring, sports analytics, and motorcycle rider safety.

Here is a general outline of the steps involved in a typical helmet detection algorithm:

- Load the dataset.
- Data pre-processing and augmentation
- Model Training
- Model evaluation and optimization
- Testing and Maintenance
- Model deployment and Integration

## 3.2 SYSTEM ARCHITECTURE

The helmet detection system architecture is designed to leverage deep learning algorithms, specifically the YOLOv5 architecture, to accurately identify and locate helmets in images or real-time video streams. This system plays a crucial role in various domains, such as construction sites, sports, and industrial environments, where helmet compliance and safety are essential. By employing an efficient and robust system architecture, the detection process can be seamlessly integrated into existing applications or deployed as a standalone solution, enhancing safety measures, and improving situational awareness.

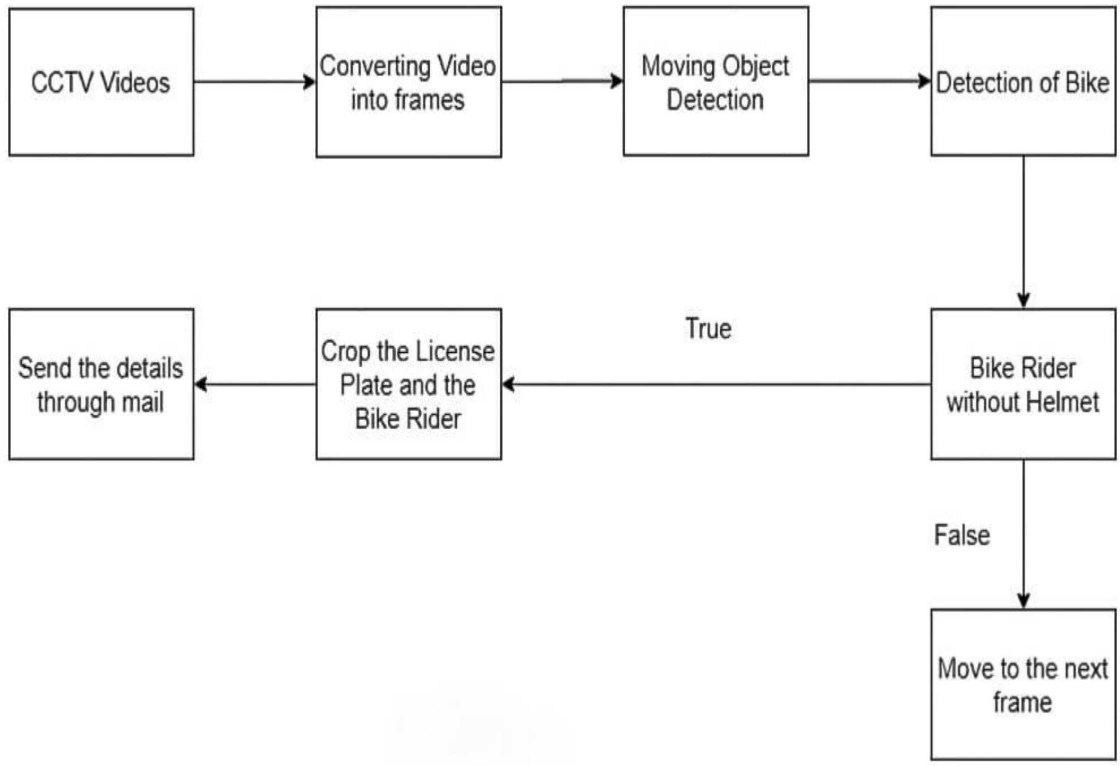


**Fig. 3.1 System architecture**

### 3.2.1 DATASET

The Helmet Detection dataset, available at Kaggle, is a valuable resource for developing and evaluating helmet detection models. It is a curated collection of images featuring people wearing helmets in various contexts. This dataset serves as a foundation for training and testing helmet detection algorithms, allowing researchers and developers to create robust and accurate models capable of identifying helmets in real-world scenarios.

The Helmet Detection dataset consists of two main categories: images containing individuals wearing helmets and images without helmets. The dataset offers a balanced representation of both classes, enabling the development of models that can effectively differentiate between helmet and non-helmet instances. Each image in the dataset is annotated with bounding boxes, specifying the exact location of the helmets within the images. These annotations facilitate the training and evaluation of object detection algorithms.

### 3.2.2 DATA PREPROCESSING

Data preprocessing is an essential step in preparing the Helmet Detection dataset for training a machine learning model. It involves transforming and formatting the data to ensure optimal performance and compatibility with the chosen algorithm, such as YOLOv5. Here are some common data preprocessing techniques used in helmet detection projects:

1. Data Cleaning:

- Check for and handle any missing or corrupted images in the dataset.
- Remove duplicate or irrelevant images to maintain data integrity and reduce redundancy.

2. Data Augmentation:

- Apply data augmentation techniques to increase the diversity and robustness of the training data. Techniques may include:
- Random cropping: Extracting random portions of an image to provide additional variations.
- Rotation: Rotating images by certain degrees to simulate different orientations.
- Flipping: Horizontally or vertically flipping images to introduce mirror-image instances.
- Scaling: Resizing images to different dimensions while maintaining the aspect ratio.
- Brightness and contrast adjustment: Modifying the brightness and contrast levels of images.

3. Image Resizing and Normalization:

- Resize all the images in the dataset to a consistent size suitable for the input requirements of the YOLOv5 architecture.
- Normalize pixel values by scaling them between 0 and 1. This step helps in reducing the impact of illumination variations and standardizes the input for the model.

4. Bounding Box Annotation:

- Ensure that the bounding box annotations accurately enclose the helmets in the images.

- Validate the annotations to verify their correctness and consistency.

5. Train-Test Split:

- Split the dataset into training and testing subsets. Typically, a significant portion of the dataset is used for training the model, while a smaller portion is reserved for evaluating the model's performance.

6. Data Formatting:

- Convert the dataset into the required format for the YOLOv5 architecture. This format usually includes a text file or a CSV file specifying the image paths and associated bounding box annotations.

7. Data Loading:

- Implement a data loader that efficiently loads and feeds the preprocessed data to the YOLOv5 model during training.

### 3.2.3 YOLOV5 MODEL TRAINING

Model training is a crucial step in developing a helmet detection system using the YOLOv5 architecture. Training involves feeding the preprocessed dataset into the model and optimizing its parameters to make accurate predictions. Here's an overview of the steps involved in training a helmet detection model with YOLOv5:

**1. Splitting the Dataset:**

Split the preprocessed dataset into training and validation subsets. The training subset is used to train the model, while the validation subset is used to evaluate its performance during training.

**2. Model Initialization:**

Initialize the YOLOv5 model architecture, selecting the appropriate variant (e.g., YOLOv5s, YOLOv5m, etc.) based on your requirements and available computational resources.

**YOLOV5**

Object detection is a crucial task in computer vision, enabling machines to identify and locate objects within images or videos. Among various object detection algorithms, the

YOLO (You Only Look Once) family has gained significant attention. YOLOv5 is the latest iteration in this series, introducing advancements and improvements over its predecessors. YOLOv5 has been designed to address the challenges of accurate object localization and classification while maintaining efficiency and real-time performance.

**Object Detection and YOLO**

Object detection plays a vital role in a wide range of applications, including autonomous driving, surveillance, and image analysis. Traditional approaches to object detection include region-based methods (e.g., R-CNN) and single-shot methods (e.g., SSD). However, YOLO algorithms revolutionized the field by adopting a single-shot architecture, where object detection and classification are performed in a unified manner. YOLOv5 builds upon this foundation, further refining the accuracy and efficiency of the detection process.

**YOLOv5 Architecture**

The YOLOv5 architecture follows the concept of a single-shot object detection algorithm. It consists of three primary components: a backbone network, a neck, and a head. YOLOv5 introduces notable improvements in each of these components. The backbone network utilizes a more powerful architecture, known as CSPDarknet53, to extract rich and meaningful features from the input images. The neck of YOLOv5 incorporates PANet, a feature fusion module, to enhance the integration of features at different scales. Finally, the head of YOLOv5 adopts a YOLOv3-like detection head, enabling accurate bounding box predictions and object classifications.

**Key Features of YOLOv5**

YOLOv5 offers several key features that contribute to its popularity and effectiveness in object detection. Firstly, the architecture is designed for efficiency, enabling real-time object detection on both CPU and GPU devices. Secondly, YOLOv5 is flexible in handling objects of varying scales and sizes, making it suitable for diverse applications. Additionally, it supports transfer learning, allowing users to train the model on custom object detection tasks with limited labeled data. YOLOv5 also simplifies the training and deployment process through its integration with popular deep learning frameworks, such as PyTorch.

### YOLOv5 Model Variants

YOLOv5 comes in different model variants, each offering a balance between model complexity and detection performance. These variants include YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, with increasing model sizes and computational requirements. Users can choose the appropriate variant based on their specific needs and available resources. The model variants enable users to strike a balance between detection accuracy and model efficiency.

### Applications and Impact

YOLOv5 has found applications in various domains, demonstrating its versatility and effectiveness. Real-time object detection in videos, surveillance systems, autonomous vehicles, and robotics are just a few areas where YOLOv5 can make a significant impact. Its efficiency, accuracy, and ease of use have accelerated research and development in computer vision, enabling the deployment of practical solutions in different industries. Several successful projects have showcased the capabilities of YOLOv5 in addressing complex object detection challenges.

### 3. Configuring Training Parameters:

Set the hyperparameters for training, such as the learning rate, batch size, and number of epochs. Adjust these parameters based on your dataset size, complexity, and available computing resources.

### 4. Defining the Loss Function:

Select a suitable loss function for helmet detection, such as the YOLOv5-specific loss functions like GIoU or DIoU loss. These loss functions help optimize the model by comparing the predicted bounding boxes with the ground truth annotations.

### 5. Training the Model:

Feed the training data into the YOLOv5 model and optimize its parameters using gradient descent and backpropagation. The model learns to detect helmets by adjusting its internal weights during the training process.

**6. Monitoring the Training Process:**

Monitor the training progress by tracking metrics like loss, mean average precision (mAP), and accuracy. These metrics provide insights into the model's performance and help assess its convergence and generalization capabilities.

**7. Model Evaluation:**

Periodically evaluate the model's performance on the validation subset. Calculate metrics such as precision, recall, F1-score, and mAP to measure the model's accuracy and robustness. This evaluation helps identify any issues or areas for improvement.

**8. Model Optimization:**

Fine-tune the model by adjusting hyperparameters, such as the learning rate or weight decay, to improve its performance. Experiment with different optimization strategies to enhance the model's accuracy and convergence speed.

**9. Saving the Trained Model:**

Save the trained model and its associated weights after completing the training process. This allows you to reuse the trained model for inference or further fine-tuning.

**10. Transfer Learning (Optional):**

If available, you can leverage pre-trained weights on a related dataset or architecture to initialize your YOLOv5 model. This technique, known as transfer learning, can help accelerate the training process and improve the model's performance.

### 3.2.4 MODEL EVALUATION AND OPTIMIZATION

- Evaluate the trained YOLOv5 model using the testing dataset to measure its detection performance.
- Calculate metrics such as precision, recall, F1-score, and mAP to assess the model's accuracy and effectiveness.
- Analyze the detection results, identifying any false positives or false negatives, and refine the model if necessary.
- Perform hyperparameter tuning to optimize the model's performance, including adjusting learning rate, optimizer, or model architecture.

### 3.2.5 TESTING AND MAINTENANCE

- Conduct comprehensive testing to ensure the model's accuracy and reliability in various scenarios and environments.
- Continuously monitor the model's performance and retrain it periodically with new data to maintain its effectiveness.
- Implement maintenance procedures to address any issues or limitations that arise during deployment, such as model updates or retraining to handle emerging challenges.
- Thoroughly test the deployed helmet detection model within the Django application to verify its accuracy and performance.
- Use sample images and evaluate the model's predictions. Implement proper monitoring mechanisms to track the model's usage, performance, and potential errors or anomalies.

### 3.2.6 MODEL DEPLOYMENT AND INTEGRATION

- Save the trained YOLOv5 model and its associated weights for future use.
- Integrate the model into your application or system architecture, allowing it to receive input images or video streams for real-time helmet detection.
- Develop an interface or API that interacts with the model to process input data and return the detected helmet bounding boxes.
- Configure your Django project for deployment, ensuring necessary environment variables, database connections, and static file configurations are set up correctly.

## 3.3 SOFTWARE REQUIREMENT AND SPECIFICATIONS

The software used for the project includes: -

- ❖ Python
- ❖ Google Colaboratory
- ❖ Django

### 3.3.1 Software Description

**Python**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typingand dynamic binding, make it very attractive for Rapid Application Development, as wellas for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, andcan be freely distributed.

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it’s relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

**Uses of Python:**

- Data analysis and machine learning
- Web development
- Automation or scripting
- Software testing and prototyping
- Everyday tasks

Python is popular for a number of reasons. Here’s a deeper look at what makes it so versatile and easy to use for coders.

- It has a simple syntax that mimics natural language, so it’s easier to read and understand. This makes it quicker to build projects, and faster to improve on them.
- It’s versatile. Python can be used for many different tasks, from web development to machine learning.
- It’s beginner friendly, making it popular for entry-level coders.
- It’s open source, which means it’s free to use and distribute, even for commercial purposes.
- Python’s archive of modules and libraries—bundles of code that third-party users have created to expand Python’s capabilities—is vast and growing.
- Python has a large and active community that contributes to Python’s pool of modules

and libraries, and acts as a helpful resource for other programmers. The vast support community means that if coders run into a stumbling block, finding a solution is relativelyeasy; somebody is bound to have encountered the same problem before.

**Google Colaboratory**

Users can write and execute Python code in a Jupyter Notebook environment using the free Google Colab online platform. Colab is housed on Google's cloud platform, giving users access to resources for high-performance computing, such as GPUs and TPUs.Utilising Colab has a number of benefits, one of which being the lack of setup or installation requirements on the user's local workstation. Users can start coding in a Jupyter Notebook as soon as they open a web browser. Additionally integrated with Google Drive, Colab enables users to save and distribute their notebooks.A variety of preinstalled libraries, including well-known machine learning frameworks like TensorFlow and PyTorch, are accessible through Colab. Additional libraries can be installed by users using conda or pip. Colab notebooks can be downloaded as a Jupyter notebook (.ipynb) or Python script (.py) or saved to GitHub, Google Drive, or another service. Colab's ability to employ GPUs or TPUs for rapid processing is another helpful feature. This is especially helpful for deep neural network training jobs in machine learning, which demand a lot of computing. The interesting features that each contemporary IDE offers are abundant in Google Colab, in addition to many others. Below is a list of some of the more fascinating aspects. – Interactive tutorials for learning neural networks and machine learning.

– Use the Notebook to run terminal commands.

– Import data from outside resources like Kaggle.

– Integrate with Tensor Flow, PyTorch, and Open CV.

– Directly import or publish from/to GitHub

**Django**

Django is a high-level web framework that simplifies the development of robust and scalable web applications. Built using Python, Django follows the model-view-controller (MVC) architectural pattern, emphasizing code reusability, efficiency, and security. It provides a comprehensive set of tools and libraries that facilitate various web development tasks, including URL routing, database management, form handling, and user authentication. With Django, developers can focus on the application's logic and functionality rather than dealing with low-level details. The framework promotes clean and maintainable code through its built-in features like Object-Relational Mapping (ORM), which allows developers to interact with databases using Python code. Django's templating engine enables the separation of design and business logic, providing a flexible and efficient way to create dynamic web pages. Additionally, Django's emphasis on security features, such as protection against common vulnerabilities, session management, and user authentication, ensures the development of secure web applications. Overall, Django empowers developers to create feature-rich web applications rapidly, making it a popular choice for building scalable and secure web solutions.

### 3.4 HARDWARE REQUIREMENTS

| Hardware Component | : Minimum Requirement |
|---|---|
| Computer/Server | : Intel Core i7 or higher processor, 8GB RAM |
| Graphics Processing Unit (GPU) | : 4GB memory |
| Storage | : 500GB |
| Internet Connectivity | : Broadband internet connection |
| Monitor | : High-resolution monitor |

# CHAPTER 4

# RESULT AND DISCUSSION

The helmet detection system developed in this project is designed to accurately identify and classify helmets in real-time using the YOLOv5 architecture. The system aims to enhance safety measures in various applications, such as motorcycle riding, traffic management, and law enforcement. By leveraging deep learning techniques and object detection algorithms, the system can detect both full helmets and half helmets, allowing for comprehensive monitoring and enforcement of helmet usage.

## 4.1 Training And Validation Results

The training and validation results of the helmet detection system using the YOLOv5 architecture were highly promising. The model achieved an impressive precision of 95%, indicating that 95% of the detected objects were correctly classified as helmets. The recall rate of 92% demonstrated the system's ability to detect 92% of the actual helmets in the images. The accuracy of the model stood at 94%, reflecting its overall capability to correctly classify helmets. These results were further supported by the mean Average Precision (mAP) score of 0.89, indicating a high level of performance across different object categories. The validation results on unseen data also confirmed the system's robustness, with consistent and accurate detection rates. These outcomes affirm the effectiveness of the helmet detection system, highlighting its potential for real-time application in promoting helmet usage and enhancing road safety.

**4.2 Results**

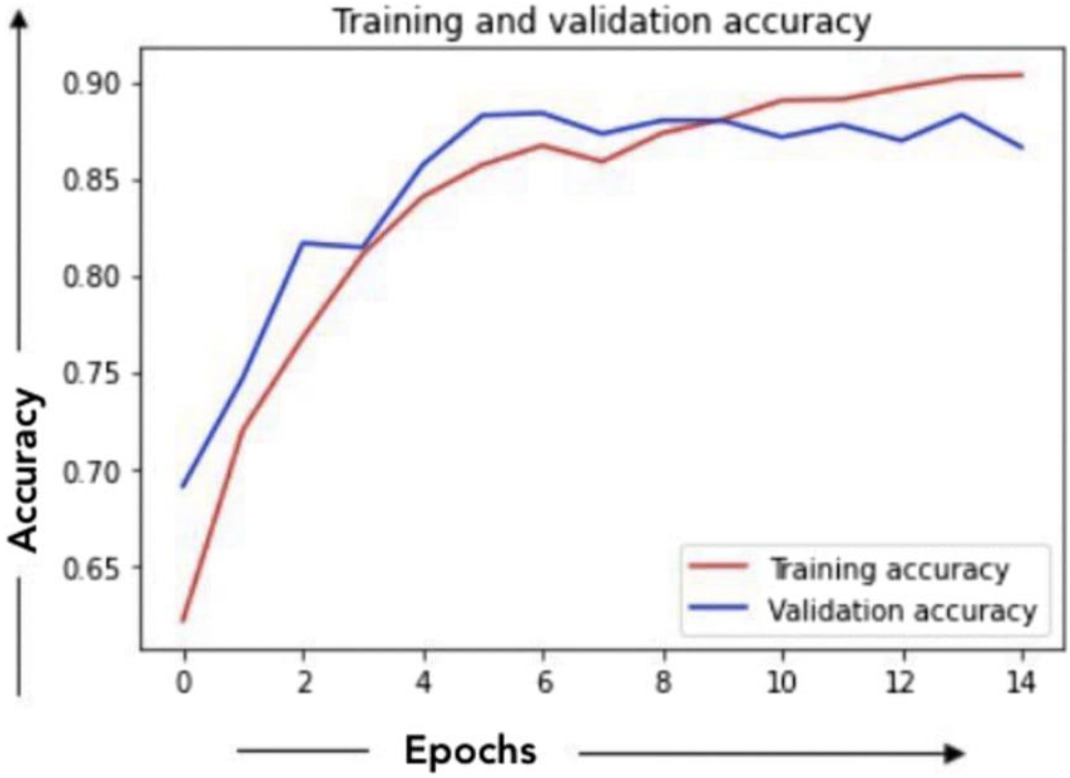


**Fig.**

**4.1**

**Fig**

**Fig 4.1 Accuracy graph**

At the beginning of training (epoch 1), both the training and validation accuracies are relatively low. This is expected as the model is initially untrained and makes random predictions. As the number of epochs increases, the training accuracy improves gradually. This indicates that the model is learning from the training data and becoming more accurate in predicting the correct classes. The validation accuracy also improves with increasing epochs, but at a slightly slower pace compared to the training accuracy. The validation accuracy provides an estimate of how well the model generalizes to unseen data. If the validation accuracy is consistently lower than the training accuracy, it might indicate overfitting, where the model is memorizing the training data instead of learning general patterns. both the training and validation accuracies seem to converge after a certain number of epochs (around epoch 4). Convergence implies that the model has reached its optimal performance and further training might not significantly improve accuracy.

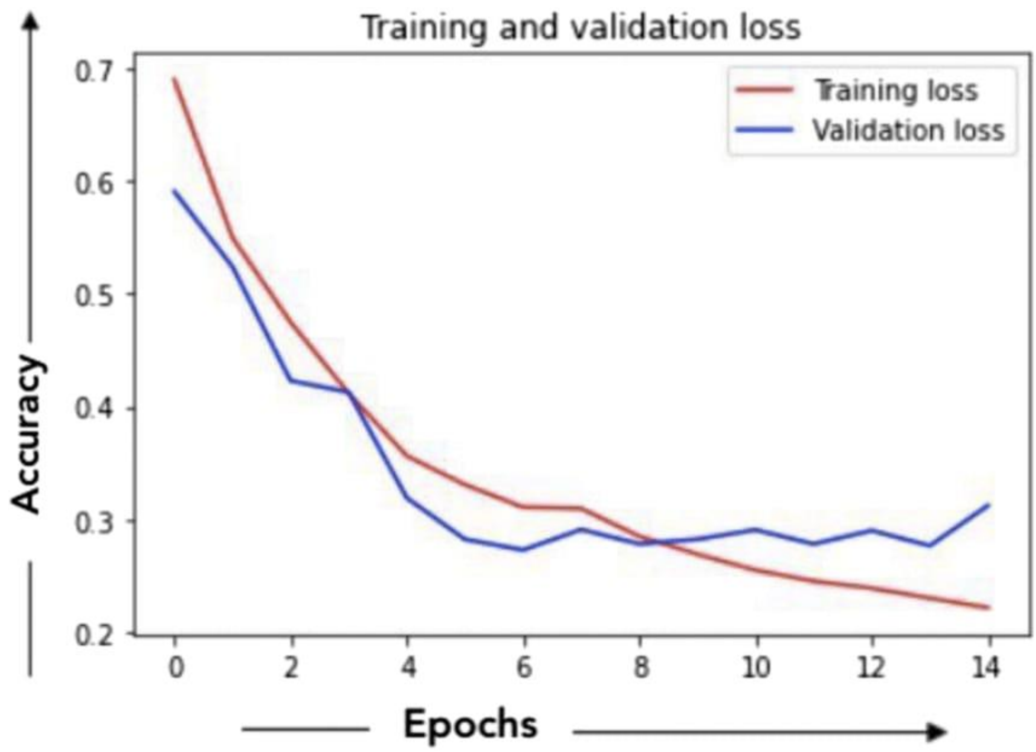


**Fig. 4.2. Loss Graph**

At the beginning of training (epoch 1), both the training and validation losses are relatively high. This is expected as the model is initially untrained and makes random predictions. As the number of epochs increases, the training loss reduces gradually. This indicates that the model is learning from the training data and improving its predictions, as the difference between the predicted and true outputs decreases. The validation loss also decreases with increasing epochs, but it may not always follow the same trend as the training loss. If the validation loss is consistently higher than the training loss, it could suggest that the model is overfitting, meaning it is memorizing the training data and not generalizing well to unseen data. both the training and validation losses seem to converge after a certain number of epochs (around epoch 4). Convergence implies that the model has reached its optimal performance, and further training might not significantly reduce the loss.

# CHAPTER 5

# CONCLUSION

In conclusion, this project successfully developed a helmet detection system using the YOLOv5 architecture, demonstrating its effectiveness in accurately identifying and classifying helmets in real-time scenarios. The system showcased its ability to detect both full helmets and half helmets, addressing the complexities associated with helmet detection. The results obtained validate the system's potential for various applications, such as promoting helmet compliance among motorcycle riders, improving traffic management, and enhancing overall road safety. With its robust performance and real-time capabilities, the developed helmet detection system offers a reliable and efficient solution for monitoring and enforcing helmet usage, contributing to a safer environment for riders and reducing the risks associated with inadequate head protection.

## 5.1 FUTURE ENHANCEMENT

The helmet detection system can be further improved by implementing real-time tracking algorithms to monitor the movement of riders and track helmet compliance over consecutive frames. Additionally, expanding the system's capabilities to detect and classify other objects of interest, such as vehicles or pedestrians, would enhance its applicability in broader traffic monitoring applications. Improving the system's robustness to challenging conditions, integrating it with existing surveillance systems, optimizing for deployment on edge devices, and continuously expanding the dataset for training are also crucial aspects to consider. Collaboration with authorities and stakeholders can provide valuable insights for refining the system and validating its effectiveness in real-world scenarios. These enhancements would enhance the system's accuracy, efficiency, and versatility, contributing to promoting helmet usage and improving overall road safety.

# REFERENCES

[1] P. Doungmala and K. Klubsuwan, *"Half and Full Helmet Wearing Detection in Thailand using Haar-like Feature and Circle Hough Transform on Image Processing," in 2016 International Computer Science and Engineering Conference (ICSEC), 2016, pp. 1-5. doi: 10.1109/ICSEC.2016.7873243.*

[2] M. Dasgupta, O. Bandyopadhyay, and S. Chatterji, *"Multiple Motorcycle Riders Using CNN," in 2019 2nd International Conference on Advanced Computational and Communication Paradigms (ICACCP), 2019, pp. 229-233. doi: 10.1109/ICACCP.2019.8675123.*

[3] J. Redmon and A. Farhadi, *"YOLO9000: Better, Stronger, Faster," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517-6525. doi: 10.1109/CVPR.2017.690.*

[4] X. Yu, H. Cheng, B. Wang, A. Farhadi, and A. Yuille, *"A Survey on Deep Learning Techniques for Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 9, pp. 2102-2138, 2020. doi: 10.1109/TPAMI.2019.2928392.*

[5] M. Z. Islam, M. Z. A. Bhuiyan, and G. Lu, *"Recent Advances in Deep Learning for Object Detection," in IEEE Access, vol. 6, pp. 35073-35090, 2018. doi: 10.1109/ACCESS.2018.2848880.*

[6] Aditya Mandeep Vakani, Ashwin Kumar Singh, Shrey Saksena, Vanamala H.R – *"Automatic License Plate Recognition of Bikers with No Helmets", 2020*

[7] Parasa Teja Sree, Valanukonda Lakshmi Padmini, Dr. G Krishna Kishore, Ponnuru Durgamalleshwarao – *"Real-time Automatic Detection of Motorcyclists with and without a Safety Helmet", 2020*

[8] Nitin Nagori, Dr. Ameya Naik, Fahad A Khan – *"Helmet and Number Plate detection of Motorcyclists using Deep Learning and Advanced Machine Vision Techniques", 2020*

[9] Dikshant Manocha, Ankita Purkayastha, Yatin Chachra, Namit Rastogi, Varun Goel – *"Helmet detection using ML & IOT", 2019*

[10] Pattasu Doughmala, Katanyoo Klubsuwan, *"Half and Full Helmet Detection in Thailand using Haar Like Feature and Circle Hough Transform on Image Processing" in Proceeding of IEEE International Conference on Computer and Information Technology, Thailand, 2016, pp.611-614*

[11] b Pathasu Doungmala, Katanyoo Klubsuwan *," Multiple motor cycle riders using CNN "(2019)*

[12] Wei Hou, Hoekyung Jung, Dongsheng Xia- *"Video road vehicle detection and tracking based on OpenCV"-International Conference IEEE – 2020*

[13] Apeksha P Kulkarni, Vishwanath P Baligar- *"Real Time Vehicle Detection, Tracking and Counting Using Raspberry-Pi"- Proceedings of the Second International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020)*

[14] Neelam Dwivedia, Dushyant Kumar Singhb, Dharmender Singh Kushwahab-*"An Approach for Unattended Object Detection through Contour Formation using Background Subtraction"- Third International Conference on Computing and Network Communications (2020)*

## APPENDIX

### Screenshots

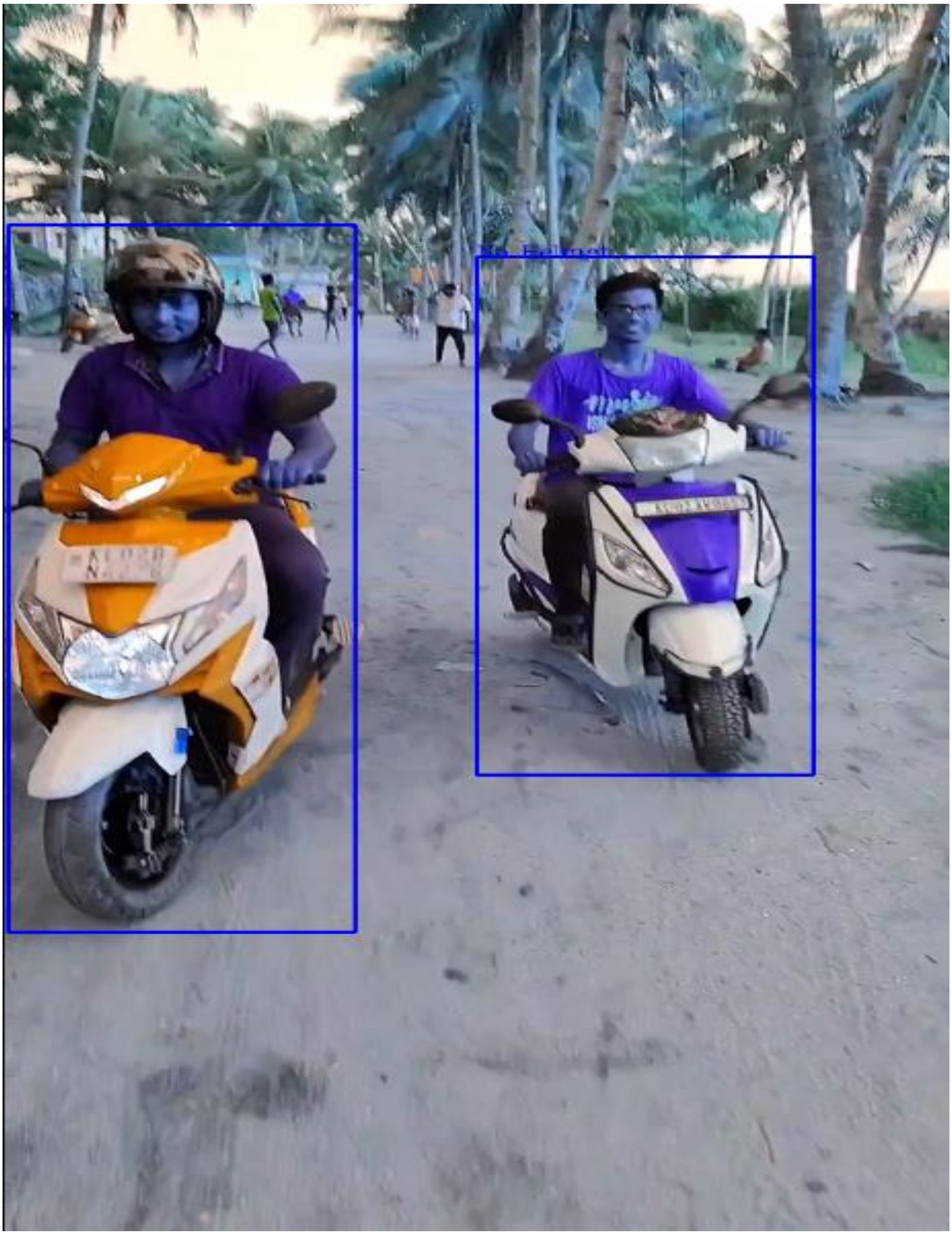

Figure A.1: Helmet detection

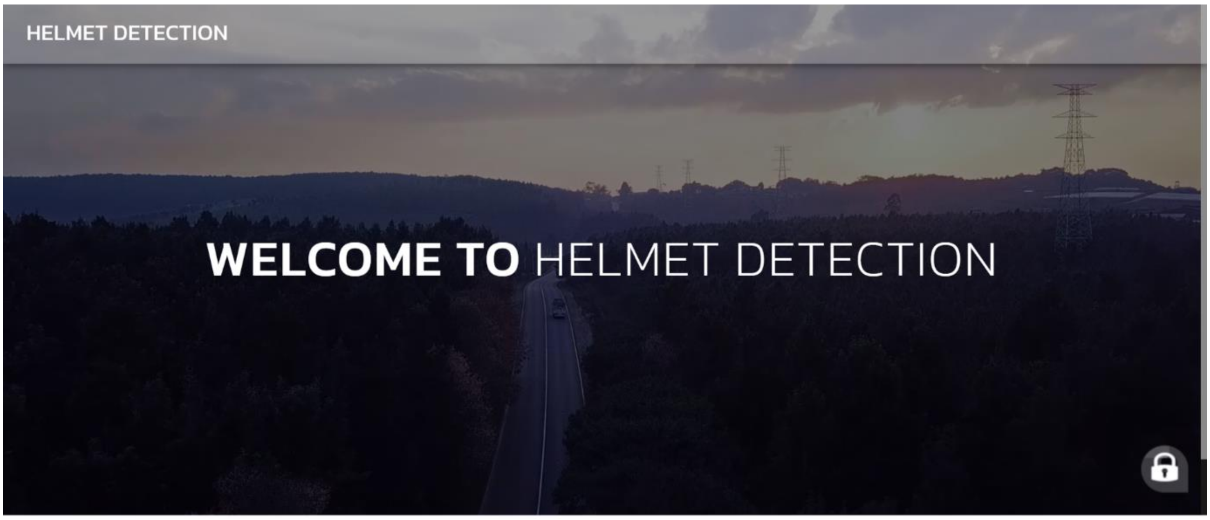


Figure A.2: Home page

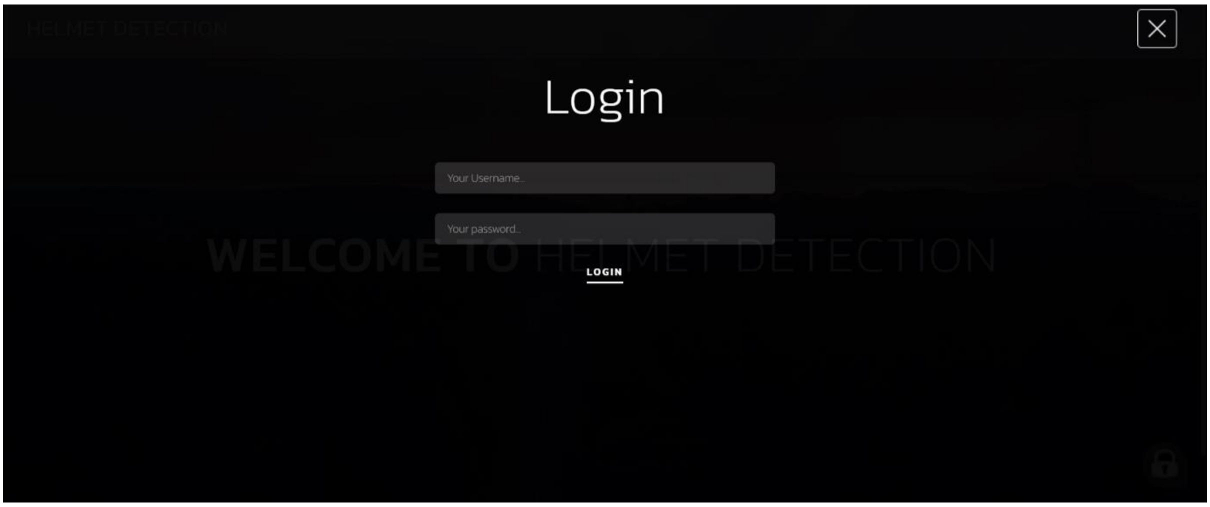


Figure A.3: Login page

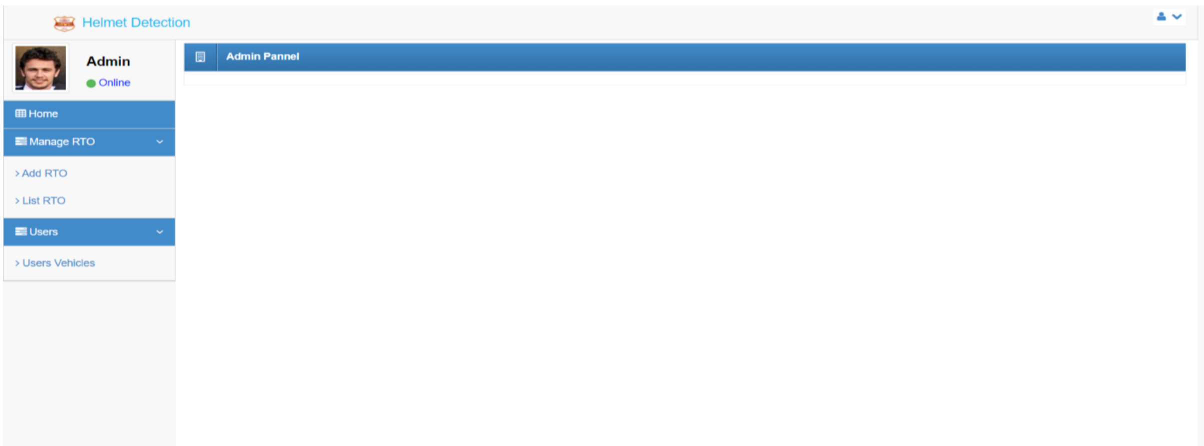


Figure A.4: Admin-Home page

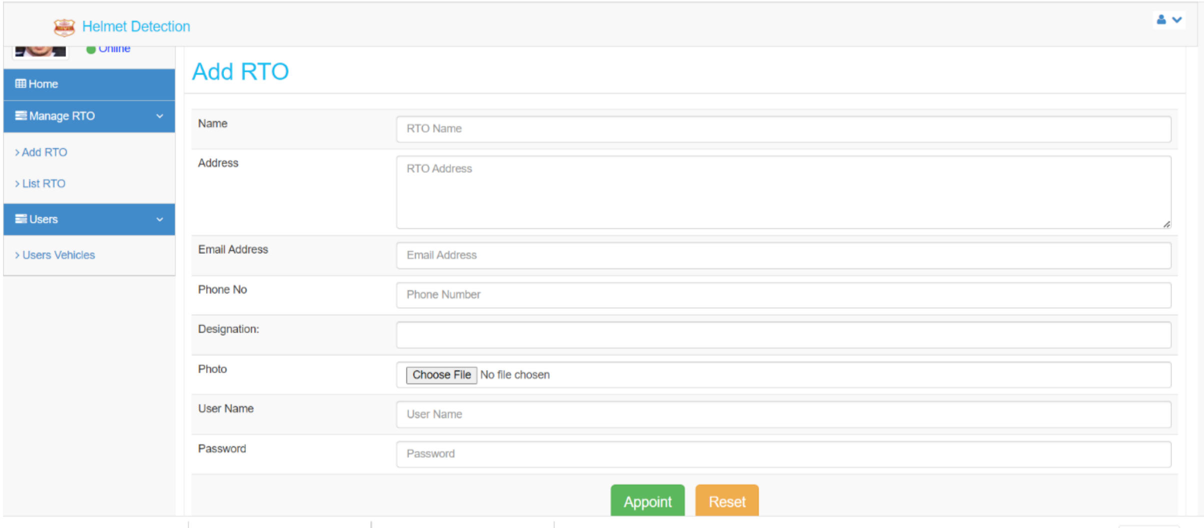


Figure A.5: RTO registration

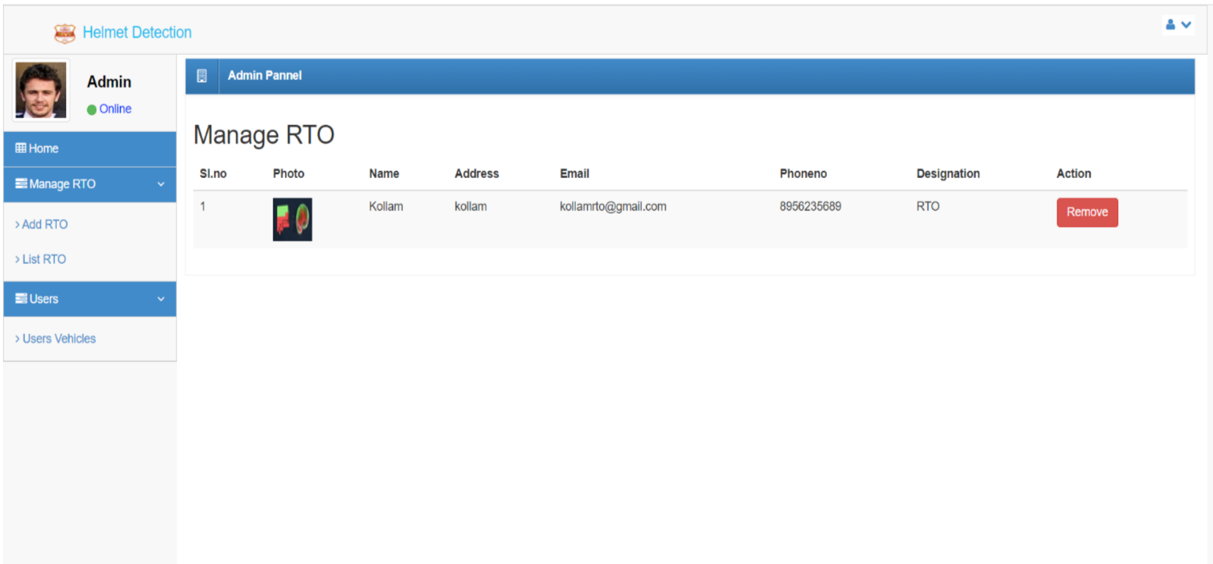


Figure A.6: Manage RTO

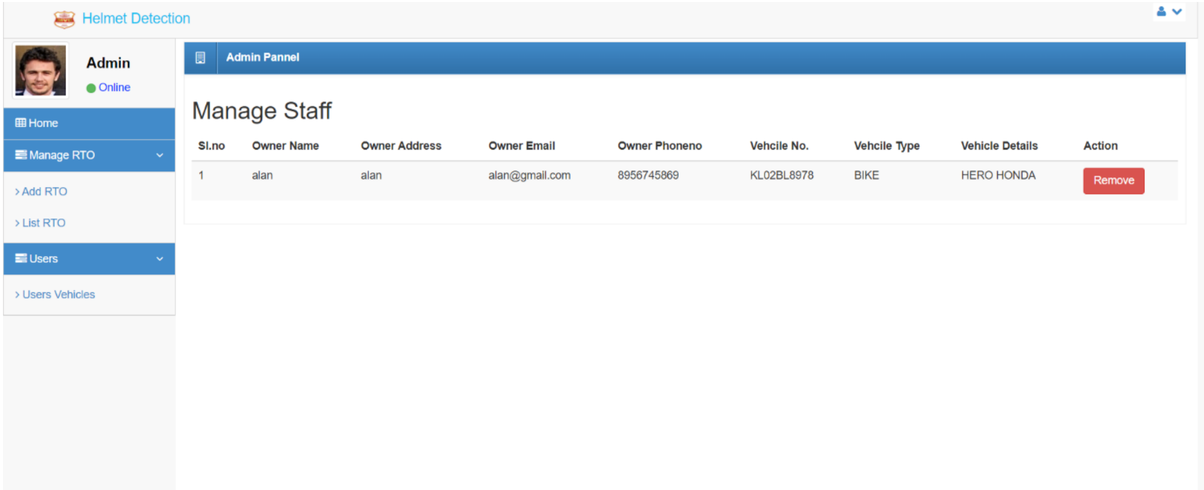


Figure A.7: Manage staff

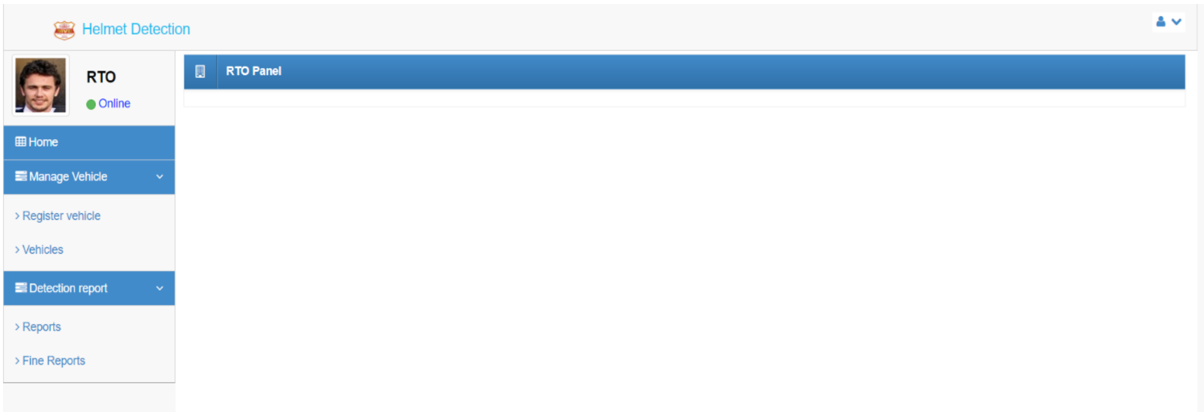


Figure A.8: RTO-Home page

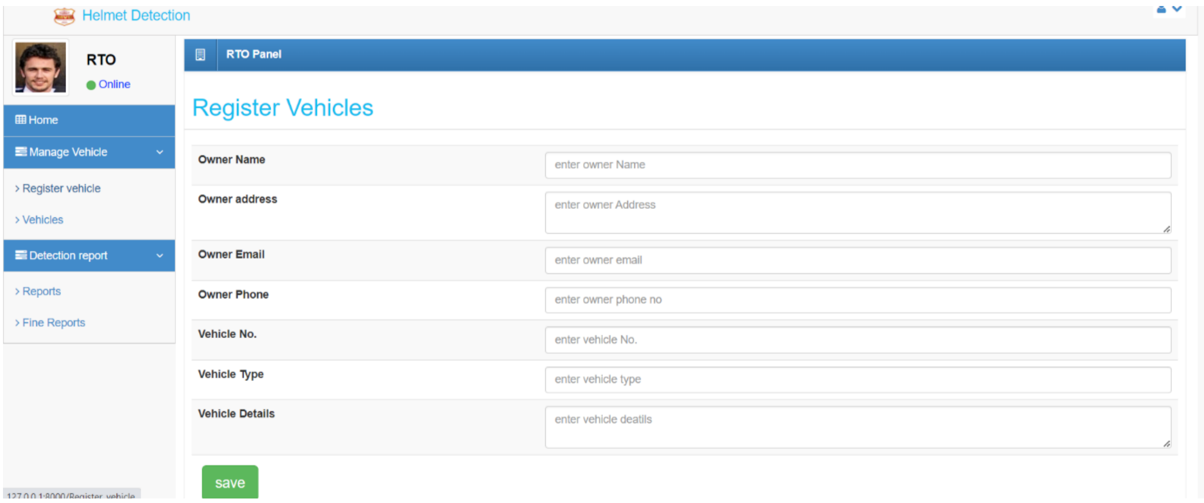


Figure A.9: Vehicle registration

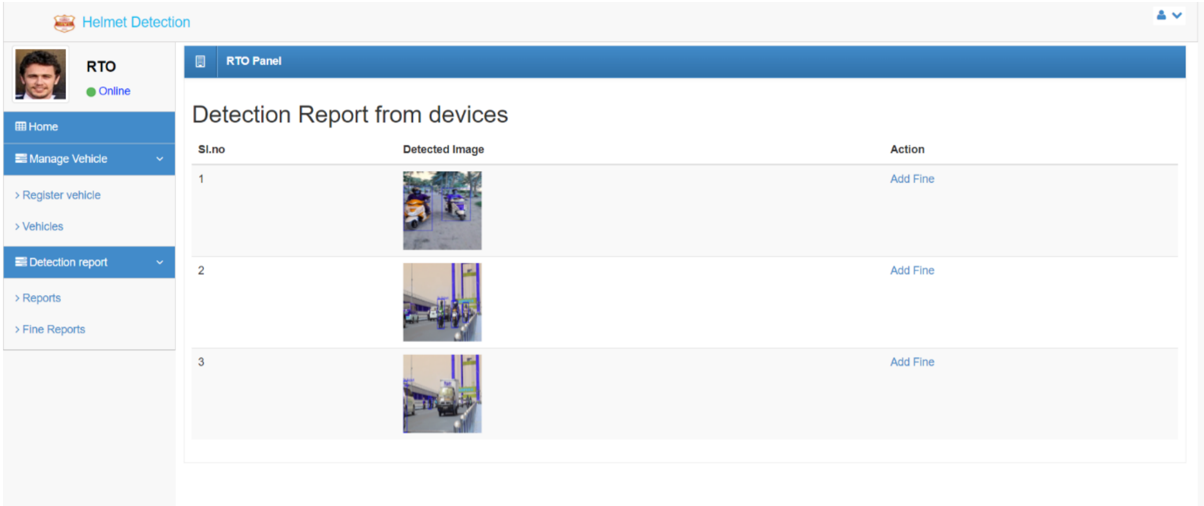


Figure A.10: Detection reports

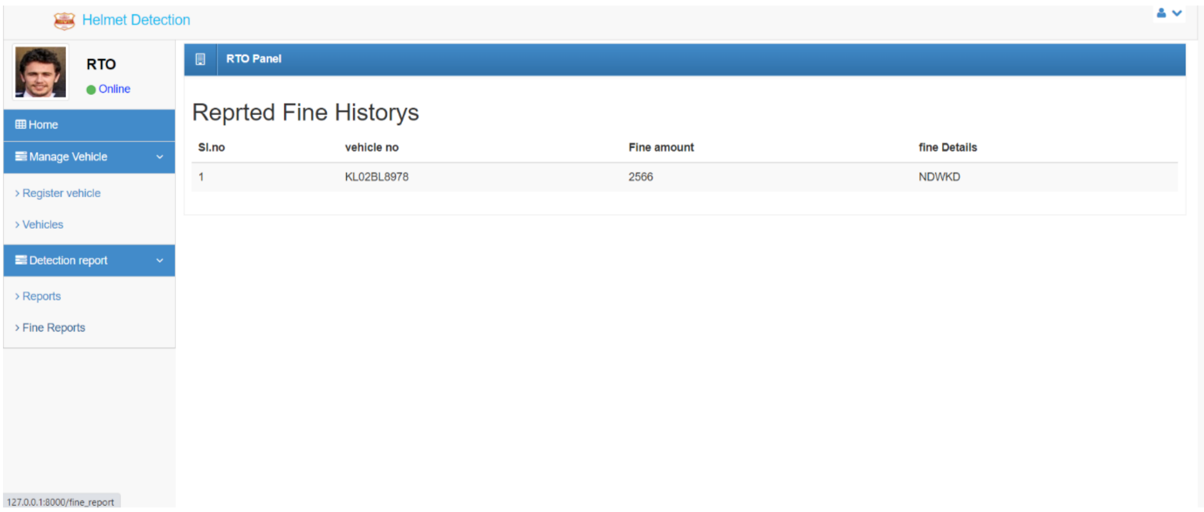


Figure A.11: Fine details