IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# TRAFFIC VIOLATION PREDICTION USING DEEP LEARNING BASED ON HELMETS WITH NUMBER PLATE RECOGNITION

***Submitted by***

**BALAJI.P (815420104009)**
**GOBINATH.G (815420104014)**
**MURUGAN.N (815420104025)**
**SUTHAGAR.S (815420104046)**

## ABSTRACT

Helmet violation detection is a crucial aspect of road safety, as it can significantly reduce the number of fatalities and injuries caused by motorcycle accidents. In recent years, computer vision techniques have been widely used to develop automated systems for helmet violation detection. This project proposes ahelmet violation detection system using image processing and machine learning techniques. The proposed system employs computer vision algorithms to detect whether a motorcyclist is wearing a helmet or not. The system is based on a deeplearning model, specifically Convolutional Neural Networks (CNN), to classify the input images into two classes, i.e., helmet and non-helmet. The system is trained on a large dataset of images with different lighting conditions, backgrounds, and helmet types to enhance its accuracy and generalization ability. The proposed system can be implemented on existing surveillance cameras installed at strategic locations on the road. This system has the potential to increase road safety and reduce the

number of motorcycle accidents caused by the violation of helmet-wearing rules. The system involves person detection, helmet, vs.no-helmet, classification using YOLO algorithm. Convolutional neural network with sequential model is implementing for number plate detectionprocess. CNN classification model proposes for classify the number plate in image and extract the user details. Then calculate the fine amount.

# CHAPTER 1

# INTRODUCTION

## 1.1 ARTIFICIAL INTELLIGENCE

AI (artificial intelligence) is the simulation of human intelligence processesby machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (usingrules to reach approximate or definite conclusions) and self-correction. Particular applications of AI include expert systems, speech recognition and machine vision. AI can be categorized in any number of ways, but here are two examples.The first classifies AI systems as either weak AI or strong AI. Weak AI, also known as narrow AI, is an AI system that is designed and trained for a particulartask. Virtual personal assistants, such as Apple's Siri, are a form of weak AI. Strong AI, also known as artificial general intelligence, is an AI system with generalized human cognitive abilities so that when presented with an unfamiliar task, it has enough intelligence to find a solution. The Turing Test, developed bymathematician Alan Turing in 1950, is a method used to determine if a computercan actually think like a human, although the method is controversial.

## 1.2 TECHNOLOGIES IN AI

- **Automation**: What makes a system or process function automatically? For example, robotic process automation (RPA) can be programmed to perform high-volume, repeatable tasks that humans normally performed. RPA is different from IT automation in that it can adapt to changing circumstances.

- **Machine learning:** The science of getting a computer to act without programming. Deep learning is a subset of machine learning that, in very simple terms, can be thought of as the automation of predictive analytics. There are three types of machine learning algorithms

- **Supervised learning:** Data sets are labeled so that patterns can be detected and used to label new data sets
- **Unsupervised learning:** Data sets aren't labeled and are sorted according to similarities or differences
- **Reinforcement learning:** Data sets aren't labeled but, after performing an action or several actions, the AI system is given feedback

• **Machine vision:** The science of allowing computers to see. This technology captures and analyzes visual information using a camera, analog-to-digital conversion and digital signal processing.

• **Natural language processing (NLP):** The processing of human -- and not computer -- language by a computer program. One of the older and best-known examples of NLP is spam detection, which looks at the subject line and the text of an email and decides if it's junk.

• **Robotics:** A field of engineering focused on the design and manufacturing of robots. Robots are often used to perform tasks that are difficult for humans to perform or perform consistently.

### 1.3 APPLICATIONS OF AI

Artificial intelligence has made its way into a number of areas. Here are sixexamples.

• **AI in healthcare:** The biggest bets are on improving patient outcomes andreducing costs. Companies are applying machine learning to make better and faster diagnoses than humans. One of the best-known healthcare technologies is IBM Watson. It understands natural language and is capable of responding to questions asked of it. The system mines patient data and other available data sources to form a hypothesis, which it then presents with a confidence scoring schema.

• **AI in business:** Robotic process automation is being applied to highly repetitive tasks normally performed by humans. Machine learning algorithmsare being integrated into analytics and CRM platforms to uncover informationon how to better serve customers. Chatbots have been incorporated into websites to provide immediate service to customers. Automation of job positions has also become a talking point among academics and IT analysts.

- **AI in education:** AI can automate grading, giving educators more time. AI can assess students and adapt to their needs, helping them work at their own pace. AI tutors can provide additional support to students, ensuring they stay.
- **AI in finance:** AI in personal finance applications, such as Mint or Turbo Tax, is disrupting financial institutions. Applications such as these collect personal data and provide financial advice. Other programs, such as IBM Watson, havebeen applied to the process of buying a home. Today, software performs much of the trading on Wall Street.
- **AI in law:** The discovery process, sifting through of documents, in law is often overwhelming for humans. Automating this process is a more efficient use of time. Startups are also building question-and-answer computer assistants that can sift programmed-to-answer questions by examining the taxonomy and ontology associated with a database.
- **AI in manufacturing:** This is an area that has been at the forefront of incorporating robots into the workflow. Industrial robots used to perform single tasks and were separated from human workers, but as the technology advanced that changed. They are used in assembly lines for car production or by NASA to move large objects in space. Researchers are also using machine learning to build robots that can interact in social settings. Computer vision, which is focused on machine-based image processing, is often conflated with machine vision.

## 1.4 DISADVANTAGES OF AI

While AI tools present a range of new functionality for businesses, artificial intelligence also raises some ethical questions. Deep learning algorithms,which underpin many of the most advanced AI tools, only know what's in the data used during training. Most available data sets for training likely contain traces of human bias. This in turn can make the AI tools biased in their function.This has been seen in the Microsoft chatbot Tay, which learned a misogynistic and anti-Semitic vocabulary from Twitter users, and the Google Photo image classification tool that classified a group of African Americans as gorillas. The application of AI in the realm of self-driving cars also raises ethical concerns. When an autonomous vehicle is involved in an accident, liability is unclear.

## 1.5 IMAGE PROCESSING

In imaging science, image processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Images are also processed as three-dimensional signals with the third-dimension being time or the z-axis. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The acquisition of images (producing the input image in thefirst place) is referred to as imaging. Closely related to image processingare computer graphics and vision. This article is about general techniques that apply to all of them. Images are also processed as three- dimensional signals with the third-dimension being time or the z-axis.

## 1.6 DEEP LEARNING

Deep learning is an artificial intelligence (AI) function that imitates the workingsof the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence

that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

- Deep learning is an AI function that mimics the workings of the human brain in processing data for use in detecting objects, recognizing speech, translating languages, and making decisions.
- Deep learning AI is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.
- Deep learning, a form of machine learning, can be used to help detect fraud or money laundering, among other functions.

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing. However,

the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incrediblepotential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support. One of the most common AI techniques used for processing big data is machine learning, a self- adaptive algorithm that gets increasingly better analysis and patterns with experience or with newly added data. This data, known simply as big data, is drawn from sources like social media, internetsearch engines, e-commerce platforms, and online cinemas, among others

## 1.7 DEEP LEARNING IN IMAGE PROCESSING

In imaging science, image processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying

standard signal-processing techniques to it. Images are also processed as three- dimensional signals with the third-dimension being time or the z-axis. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The acquisition of images (producing the input image in thefirst place) is referred to as imaging. Closely related to image processingare computer graphics and computer vision. In computer graphics, images are manually made from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from natural scenes, as in most animated movies. Computer vision, on the other hand, is often considered high-level image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans).In modern sciences and technologies, images also gain much broader scopes dueto the ever-growing importance of scientific visualization (of often large-scale complex scientific/experimental data). Examples include microarray data in genetic research, or real-time multi-asset portfolio trading in finance. Image analysis is the extraction of meaningful information from images; mainly from digital images by means of digital image processing techniques. Image analysis tasks can be as simple as reading bar coded tags or as sophisticated

as identifyinga person from their face.

Computers are indispensable for the analysis of large amounts of data, for tasks that require complex computation, or for the extraction of quantitative information. On the other hand, the human visual cortex is an excellent image analysis apparatus, especially for extracting higher-level information, and for many applications — including medicine, security, and remote sensing — human analysts still cannot be replaced by computers.

## 1.8 ADVANTAGES FOR ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) offers numerous advantages across a wide rangeof industries and applications. Some of the key advantages of AI include:

**Increased Efficiency:** AI can automate tedious and repetitive tasks, allowing humans to focus on more complex and creative work. This can lead to significant increases in productivity and efficiency.

**Improved Accuracy:** AI systems can process and analyze vast amounts of data, enabling them to make highly accurate predictions and decisions. This can be particularly useful in fields such as healthcare, where AI can help diagnose diseases and develop personalized treatment plans.

**Enhanced Personalization:** AI can be used to analyze large datasets to identify patterns and preferences, enabling personalized experiences in areas such as e- commerce, entertainment, and social media.

**Cost Savings:** AI can help reduce costs by automating tasks that would otherwiserequire human labor, such as customer service or data entry.

**Improved Safety:** AI can be used to monitor and detect potential safety issues in real-time, such as in autonomous vehicles or industrial settings.

**Increased Accessibility:** AI can enable greater accessibility for individuals with disabilities by providing assistive technologies such as speech recognition or computer vision.

# CHAPTER 2

# SYSTEM ANALYSIS

## 2.1 LITERATURE SURVEY

### 2.1.1 TITLE: AN EFFICIENT SAFETY AND AUTHORIZED HELMET DETECTION USING DEEP LEARNING APPROACH

**AUTHOR:** SUSA, JULIE ANN B

**YEAR:** 2022

This paper proposed vision-based system could aid in identifying riders who are wearing helmets that aren't fitting properly. The system used the YOLOv3 model and a deep learning approach to use the helmet detection model in this work. Creating multiple checkpoints to monitor riders for helmets may cause traffic congestion on many public roadways. It is necessary to establish anearly detection system for their safety to prevent accidents and to follow the lawof wearing authorized helmets. The YOLOv3 approach will be employed in thisresearch. The system scored 90 % above for all of the testing features as a consequence of testing and deployment. Motorcycle riding has long been considered one of the most practical modes of transportation. However, in recentyears, the number of motorcycle riders killed in traffic accidents has climbed dramatically in large cities and on highways.

**ADVANTAGES**

- Classified helmet or non-helmet.
- Reduced Human Effort.

**DISADVANTAGES**

- Not support real time data.
- Maintenance.

### 2.1.2 TITLE:CNN-BASED AUTOMATIC HELMET VIOLATION DETECTION OF MOTORCYCLISTS FOR AN INTELLIGENT TRANSPORTATION SYSTEM

**AUTHOR:** WARIS, TASBEEHA**YEAR:**2022

This paper developed a system for automatically detecting bikers without a helmet using a faster region-based convolutional neural network (R-CNN). (e system takes input in the form of video and converts that into frames to perform helmet violation detection. (e dataset has been collected from two sources, i.e., online repositories and self-captured videos

from different locations in Lahore, Pakistan. (e experimental analysis shows that the proposed system has 97.69% accuracy. It may help to take necessary actions against traffic rule violators. (is work may be extended to incorporate more features, like number plate detectionand other traffic violations, in the future. An intelligent transportation system (ITS) is an advanced application that supports multiple transport and traffic management modes. ITS services include calling for emergency rescue and monitoring traffic laws with the help of roadside units. It is observed that many people lose their lives in motorbike accidents mainly due to not wearing helmets.Automatic helmet violation detection of motorcyclists from real-time videos is ademanding application in ITS. It enables one to spot and penalize bikers withouta helmet.

**ADVANTAGES**

- Takes video as an input and performs helmet violation detection.
- Reduced Human Error.

**DISADVANTAGES**

- Computational complexity.
- Privacy Concerns.

**2.1.3 TITLE: ROBUST AUTOMATIC MOTORCYCLE HELMET VIOLATION DETECTION FOR AN INTELLIGENT TRANSPORTATION SYSTEM**

**AUTHOR:** TRAN DUONG, NGUYEN-NGOC **YEAR:** 2023

This paper presents a framework to detect and identify separate motorcycles while tracking rider specific helmet use. The helmet-use classification approach shows increased efficiency compared to previous studies.This approach builds on and extends a previous framework, taking advantage ofnew techniques and algorithms to achieve even greater accuracy and precision. By incorporating state-of-the-art technologies and best practices, we have createda powerful tool for detecting helmet use in various scenarios and settings. This approach will prove to be an invaluable resource for promoting motorcycle safety and reducing the number of preventable injuries and fatalities. The implementation of automated detection systems for motorcycle helmet usage through video surveillance has the potential to significantly enhance the efficacyof educational and enforcement campaigns, thereby contributing to increased road safety. Despite the potential benefits of these systems, several

aspects of existing detection approach still require improvement.

**ADVANTAGES**

- Differentiating between helmet-wearing drivers and passengers.
- Improved Road Safety.

**DISADVANTAGES**

- Support only image datasets.
- Cost Of Implementation.

**2.1.4 TITLE:A LIGHTWEIGHT YOLOV3 ALGORITHM USED FORSAFETY HELMET DETECTION**

**AUTHOR:** DENG, LIXIA**YEAR:**

2022

This paper proposes a lightweight object detection network: ML-YOLOv3. In this paper, three network improvement methods are proposed, which can significantly reduce the computational cost of the model while maintaining a strong detection effect. Based on the helmet dataset, CSP-Ghost-ResNet proposed by us effectively reduces the complexity of the model and achieves almost the same level of detection effect as YOLOv3. ML-Darknet reduces the detection effect of the model, but it effectively reduces the computational cost of the model. In addition, PAN-CGR-Network is redesigned in this paper. It further reduces computing costs. Because of the emergence of image processors with powerful computing power and large-scale data samples, deep learning has developed rapidly. Ishak Pacal et al.5 proposed the YOLOv3 algorithm for robust real-timepolyp detection, which effectively improves the detection effect. Yizhou Chen et al.6 systematically explained the application of generative adversarial networks in medical image augmentation. Qiu Guan et al.7 applied generative adversarial networks to medical image detection to solve the problem of insufficient data samples.

**ADVANTAGES**

- Lightweight object detection network.
- Easy to Deployment.

**DISADVANTAGES**

- Need large datasets for training.
- Training Data Requirements.

### 2.1.5 TITLE: RESEARCH ON SAFETY HELMET DETECTIONALGORITHM BASED ON IMPROVED YOLOV5S

**AUTHOR:** AN, QING**YEAR:**

2023

This paper proposed a modified YOLOv5 network to adaptively adjust theanchor box to increase the matching degree between the anchor box and the target box, which can extract discriminative image features from small targets. In the proposed method, the GAM attention mechanism is combined with the CPS module of the CBAM attention mechanism. It is added to the backbone network(Backbone) and neck network (Neck) of the original YOLOv5s network to improve the performance of the neural network by reducing the loss of feature information and amplifying the global interaction. This article introduces a three-dimensionally arranged channel attention and convolutional spatial attention sub-module with a multi-layer perceptron, and the feature map adaptively refines each convolutional block of the network structure through a combination module, which is conducive to the establishment of high dimensional spatial feature correlation and the extraction of effective features of the target. Furthermore, theCBAM is integrated into the CSP module to improve target feature extraction while minimizing computation for model operation.

**ADVANTAGES**

- Effectively improves the speed.
- High Accuracy.

**DISADVANTAGES**

- Time complexity can be high.
- Model Complexity.

## 2.1.6 TITLE: DEEP LEARNING-BASED AUTOMATIC SAFETYHELMET DETECTION SYSTEM FOR CONSTRUCTION SAFETY

**AUTHOR:** HAYAT, AHATSHAM**YEAR:** 2022

This paper used different versions of YOLO architecture, YOLOv3, YOLOv4, and YOLOv5x, to detect safety helmets due to their proven accuracy in object detection tasks. Among them, YOLOv5x achieved the best mAP (92.44%) in detecting smaller objects and objects in low-light images, thereby showing its efficacy in safety helmet detection. Despite the significant outcomesachieved by YOLOv5x-based architecture, it also possesses several limitations. The proposed deep learning model struggled to perform in some scenarios (e.g., with an obstacle in front of helmets, and objects identical to helmets). Training the model with more images, including the above-mentioned scenarios, could potentially increase the model's efficacy. Moreover, in the future, more safety tools could be added for detection, such as vests, gloves, and glasses, to ensure greater safety for workers. This literature review shows that many studies have been conducted on safety helmet detection on construction sites based on sensor,machine learning, and deep learning techniques. Most of them fail in detecting safety helmets in complex scenarios, such as sites with multiple workers.

**ADVANTAGES**

- Automatic detection of safety helmets.
- Reduced Workplace Accidents.

**DISADVANTAGES**

- Image based analysis.
- Computational Resources.

## 2.1.7 TITLE: MULTI-SCALE SAFETY HELMET DETECTION BASEDON RSSE-YOLOV3

**AUTHOR:** SONG, HONGRU**YEAR:** 2022

With the increasing acceleration of urbanization, which has led to the rapid development of infrastructure construction, safe production is an importantguarantee to promote growth. Safety helmets play an important role in ensuring safe production and can effectively protect the head safety of on-site constructionworkers, reducing the operational risk of construction workers to a certain extent. In the safe production specifications of

construction and heavy industry, it is stipulated that a worker is not allowed to enter the construction site without wearing a safety helmet. However, in practice, construction employees often failto wear safety helmets. To monitor and correct unsafe behavior and ensure the safety of construction workers, it is necessary to carry out real-time detection on whether construction workers are wearing safety helmets. On the construction site, there is often a situation where the construction personnel are far away from the monitoring equipment, which makes the monitoring target exhibit the characteristics of small-scale targets. Due to the few small-scale target pixels, inconspicuous image features, dense targets, and other reasons, small target missed detection and target occlusion is prone to occur in the detection. proposea parallel RSSE network module to replace the Res8 module in Darknet53.

**ADVANTAGES**

- Improving the detection accuracy.
- Enhanced Robustness.

**DISADVANTAGES**

- Mis-classification error rate can be occurred.
- Fine tuning and optimization.

**2.1.8 TITLE: LIGHTWEIGHT HELMET DETECTION ALGORITHMUSING AN IMPROVED YOLOV4**

**AUTHOR:** CHEN, JUNHUA**YEAR:**
2023

This paper proposes an improved YOLOv4 algorithm to achieve real-timeand efficient safety helmet wearing detection. The improved YOLOv4 algorithmadopts a lightweight network PP-LCNet as the backbone network and uses deepwise separable convolution to decrease the model parameters. Besides, thecoordinate attention mechanism module is embedded in the three output feature layers of the backbone network to enhance the feature information, and an improved feature fusion structure is designed to fuse the target information. In terms of the loss function, we use a new SIoU loss function that fuses directionalinformation to increase detection precision. However, accidents arising from workers not wearing safety helmets can be seen everywhere due to a lack of a particular sense of safety protection. Therefore, monitoring whether workers arewearing helmets is crucial to their safety. Traditional helmet inspection mainly consists of monitoring in the

surveillance room and manual patrol at the construction site. The former requires inspectors to stare at the screen for long periods, which can cause eye fatigue and lead to misjudgements and missed inspections, while the latter requires a lot of time and labour.

**ADVANTAGES**

- Reinforce feature fusion and acquire effective features.
- Robustness to Environmental Variations.

**DISADVANTAGES**

- Small images are used.
- Generalization to new environment.

**2.1.9 TITLE: VIDEO ANALYTICS FOR DETECTING MOTORCYCLISTHELMET RULE VIOLATIONS**

**AUTHOR:** TSAI, CHUN-MING**YEAR:**
2023

This paper presents two new deep learning models, YOLOv7-CBAM andYOLOv7-SimAM, which incorporate YOLOv7-E6E, CBAM, and SimAM. TheYOLOv7- E6E model was trained on images of size 1920, while the YOLOv7- CBAM and YOLOv7-SimAM models were trained on images of size 1280. These models were employed to detect the test images in Track 5, and the resultswere submitted to the AI City CHALLENGE Track 5 evaluation system. The experimental results on the 100 test videos of the 2023 AI City CHALLENGE Track 5 demonstrate the effectiveness of our methods, with mAP scores of 0.6112, 0.6389, and 0.6422 for YOLOv7-E6E, YOLOv7-CBAM, and YOLOv7-SimAM, respectively. presented two new deep learning models, YOLOv7- CBAM and YOLOv7-SimAM, which incorporate YOLOv7-E6E, CBAM, and SimAM. The YOLOv7-E6E model was trained on images of size 1920, while the YOLOv7-CBAM and YOLOv7-SimAM models were trained on images of size 1280. These models were employed to detect the test images in Track 5, andthe results were submitted to the AI City CHALLENGE Track 5 evaluation system.

**ADVANTAGES**

- Improving traffic safety measures.
- Data Insights.

**DISADVANTAGES**

- Does not support publicly available datasets
- Environmental Conditions.

**2.1.10 TITLE: REAL-TIME MULTI-CLASS HELMET VIOLATION DETECTION USING FEW-SHOT DATA SAMPLING TECHNIQUE AND YOLOV8**

**AUTHOR:** ABOAH, ARMSTRONG**YEAR:**
2023

This paper proposed and implemented a novel data processing strategy, called the "few-shot data sampling technique", for developing a robust helmet detection model with fewer annotations. The technique involves selecting a smallbut representative number of images from a large dataset using our developed algorithms and then applying data augmentation techniques to generate additional images for training. By using this technique, we able to develop a robust helmet detection model with fewer annotations, which is an important contribution as itreduces the time and effort required for annotation. Developed a real-time helmetdetection system that is robust to varying weather conditions and time of day by utilizing YOLOv8, a state-of-the-art object detection model, and data augmentation techniques. YOLOv8 is designed to be fast and accurate, making itideal for use in real time. In addition, several data augmentation strategies were utilized in this study to overcome the issues of occlusion and viewpoint problems. To further improve prediction accuracy and confidence during inference time, the study employed test time augmentation (TTA) during its inference stage.

**ADVANTAGES**

- Single-stage object detection model.
- Real Time Alerts.

**DISADVANTAGES**

- Support only video dataset.
- Data Storage and Processing.

**2.2 EXISTING SYSTEM**

Nowadays, road accidents are one of the major causes that lead to human death. Motorbike accidents can cause severe injuries. The helmet is important for every

motorcyclist. However, many fail to conform to the law of wearing helmets. Helmets are essential equipment's to protect workers from danger during inspection and operation. Considering that some workers would not always obeythe regulation, video surveillance systems covering the whole factory and supervisors are needed to monitor whether workers are wearing helmets or not. However, with a large number of surveillance screens, it is difficult to identify any helmet violation behaviour during any time, which can lead to severe accidents. With the rapid development of image recognition technologies, computer vision-based inspections have been one of the most important industrial application areas. In this paper, an intelligent vision-based approach for helmet identification is proposed. This approach focuses on monitoring whether workers are wearing helmets or not, at the same time, identifying the colors of helmets. Acolor-based hybrid descriptor composed of local binary patterns (LBP), humoment invariants (HMI) and color histograms (CH) is proposed to extract features of helmets with different colors (red, yellow and blue). Then a hierarchical support vector machine (H-SVM) is constructed to classify all features into four classes (red-helmet, yellow-helmet, blue helmet and non- helmet). The helmet is important for every motorcyclist. However, many fail to conform to the law of wearing helmets. Helmets are essential equipment's to protect workers from danger during inspection and operation.

**DISADVANTAGES**

- Only support the image datasets
- Accuracy is less
- Need to train the more datasets for helmet detection
- Detect the number plate from images2.3 PROPOSED SYSTEM

There is no automated existing system which can detect motorcyclists whoare not wearing helmets as well as masks due to which the traffic policemen haveto manually keep records of such traffic rules violators either by remembering thenumber plate or by capturing a photo of the number plate. This manual administration can sometimes lead to errors. In order to overcome these drawbacks, we have designed an automated helmet and face mask detection system which is able to catch all the motorcyclists who are not wearing helmets and masks just by storing the number plate of those bike riders. To overcome the disadvantages of the existing system, we have proposed an automated system which is more accurate and requires minimum human efforts. In this project, implement the framework to identify the helmet traffic violations in real time environments. We can use object detection

and recognition system to identify thehelmet using YOLO algorithm. And also recognize the number plate using Convolutional neural network algorithm to extract user details. The proposed approach is able to detect the object in different illumination and occlusion. The system extracts objects class based on feature extracted. The system uses You Only Look Once (YOLO)-Darknet deep learning framework which consists of Convolutional Neural Networks trained on Common Objects in Context (COCO)and combined with computer vision. And also recognize the number plate using Convolutional neural network algorithm to extract user details. Based on CNN, we can detect the text object and recognize the objects to print the number plate. The Send fine amount as SMS to non-wearing helmet persons.

**ADVANTAGES**

- Implement in real time environments
- Accuracy is high
- Recognize the numbers in captured image
- SMS alert system

# CHAPTER 3

# SYSTEM ARCHITECTURE

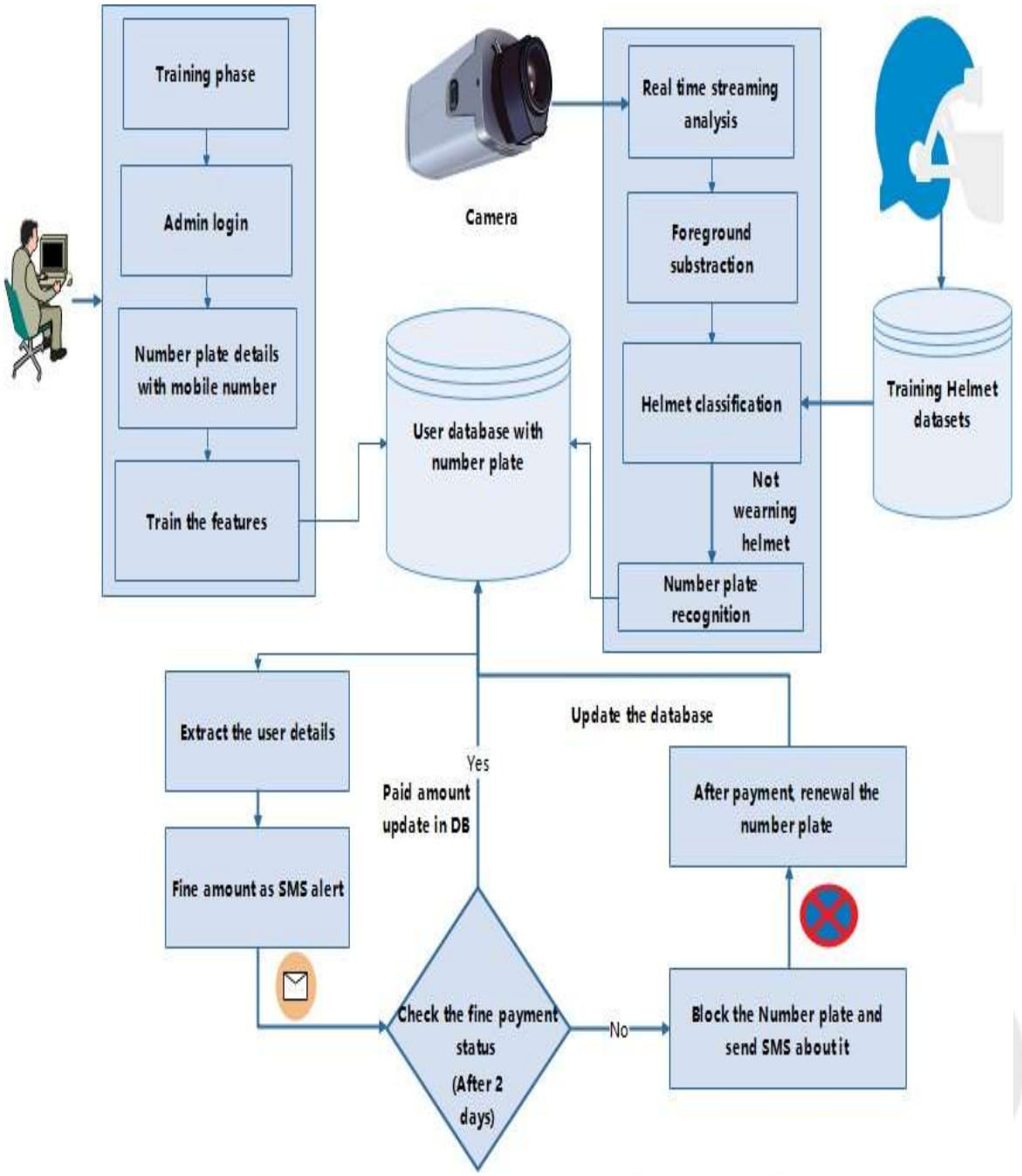


**Fig.3.1. System Architecture**

# CHAPTER 4
# SYSTEM SPECIFICATION

## 4.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design.

- Processor : Intel processor 2.6.0 GHZ
    - RAM : 16 GB
- Hard disk : 1 TB
- Compact Disk : 650 MB
- Keyboard : Standard keyboard
- Monitor : 15-inch colour monitor

## 4.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is useful in estimating cost, planning team activities and performing tasks throughout the development activity.

- Operating System : Windows OS
- Front End : PYTHON
- IDE : PYCHARM
- Back End : MYSQL
- Application : WEB APPLICATION

# CHAPTER 5

# SOFTWARE DESCRIPTION

## 5.1 PYTHON

Python is a high-level, interpreted programming language that is widely used in various domains such as web development, scientific computing, data analysis, artificial intelligence, machine learning, and more. It was first released in 1991 by Guido van Rossum and has since become one of the most popular programming languages due to its simplicity, readability, and versatility. One ofthe key features of Python is its easy-to-learn syntax, which makes it accessible to both novice and experienced programmers. It has a large standard library thatprovides a wide range of modules for tasks such as file I/O, networking, regular expressions, and more. Python also has a large and active community of developers who contribute to open-source libraries and packages that extend its capabilities. Python is an interpreted language, which means that it is executed line-by-line by an interpreter rather than compiled into machine code like C or C++. This allows for rapid development and testing, as well as easier debugging and maintenance of code. Python is used for a variety of applications, including web development frameworks such as Django and Flask, scientific computing libraries such as NumPy and Pandas, and machine learning libraries such as TensorFlow and PyTorch. It is also commonly used for scripting and automationtasks due to its ease of use and readability. Overall, Python is a powerful and versatile programming language that is widely used in a variety of domains due to its simplicity, ease of use, and active community.

Python is an interpreted high-level programming language for general- purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a largeand comprehensive standard library. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open- source software and has a community-based development model, as do nearly allof Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation. Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact

modularity has made it particularly popular as a means of adding programmableinterfaces to existing applications. Van Rossum's vision of a small core languagewith a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach. While offering choice in coding methodology, the Python philosophy rejects exuberant syntax (such as that of Perl) in favor of a simpler, less-cluttered grammar. As Alex Martelli put it: "To describe something as 'clever' is not considered a complimentin the Python culture. “Python’s philosophy rejects the Perl "there is more than one way to do it" approach to language design in favour of "there should be one—and preferably only one—obvious way to do it".

Python has a wide range of applications, including:

**Data Science:** Python is one of the most popular languages for data science, thanks to libraries like NumPy, Pandas, and Matplotlib that make it easy to manipulate and visualize data.

**Machine Learning:** Python is also widely used in machine learning and artificial intelligence, with libraries like TensorFlow, Keras, and Scikit-learn that providepowerful tools for building and training machine learning models.

**Web Development:** Python is commonly used in web development, with frameworks like Django and Flask that make it easy to build web applications and APIs.

**Scientific Computing:** Python is used extensively in scientific computing, with libraries like SciPy and SymPy that provide powerful tools for numerical analysisand symbolic mathematics.

There are two attributes that make development time in Python faster than in other programming languages:

1. Python is an interpreted language, which precludes the need to compile code before executing a program because Python does the compilation in the background. Because Python is a high-level programming language, it abstracts many sophisticated details from the programming code. Python focuses so much on this abstraction that its code can be understood by most novice programmers.

Python code tends to be shorter than comparable codes. Although Python offers fast development times, it lags slightly in terms of execution time. Compared to fully compiling

languages like C and C++, Python programs execute slower. Of course, with the processing speeds of computers these days, the speed differences are usually only observed in benchmarking tests, not in real-world operations. Inmost cases, Python is already included in Linux distributions and Mac OS X machines.

One of the strengths of Python is its rich ecosystem of third-party librariesand tools. These libraries provide a wide range of functionality, from scientific computing and data analysis to web development and machine learning. Some popular Python libraries and frameworks include:

**NumPy:** a library for numerical computing in Python, providing support for large, multi-dimensional arrays and matrices, along with a large collection of mathematical functions to operate on these arrays.

**Pandas:** a library for data manipulation and analysis in Python, providing support for reading and writing data in a variety of formats, as well as powerful tools formanipulating and analyzing data.

**Matplotlib:** a plotting library for Python that provides a variety of visualization tools, including line plots, scatter plots, bar plots, and more.

**TensorFlow:** an open-source machine learning library for Python that provides avariety of tools and algorithms for building and training machine learning models.

**Django:** a popular web framework for Python that provides a full-stack framework for building web applications, with support for everything from URL routing to user authentication and database integration.

## 5.2 TENSORFLOW LIBARIES IN PYTHON

TensorFlow is an open-source machine learning framework developed byGoogle Brain Team. It is one of the most popular libraries for building and training machine learning models, especially deep neural networks. TensorFlowallows developers to build complex models with ease, including image and speech recognition, natural language processing, and more. One of the key features of TensorFlow is its ability to handle large-scale datasets and complex computations, making it suitable for training deep neural networks. It allows forparallelization of computations across multiple CPUs or GPUs, allowing for faster training times. TensorFlow also provides a high-level API called Kera's

thatsimplifies the process of building and training models. TensorFlow offers a widerange of tools and libraries that make it easy to integrate with other Python libraries and frameworks. It has built-in support for data preprocessing and visualization, making it easy to prepare data for training and analyze model performance. One of the major advantages of TensorFlow is its ability to deploymodels to a variety of platforms, including mobile devices and the web.

**Graph-Based Computation:** TensorFlow uses a graph-based computation model, which allows for efficient execution of computations across multiple devices andCPUs/GPUs.

**Automatic Differentiation:** TensorFlow provides automatic differentiation, whichallows for efficient computation of gradients for use in backpropagation algorithms.

**High-Level Apis:** TensorFlow provides high-level APIs, such as Kera's, that allow developers to quickly build and train complex models with minimal code.

**Preprocessing And Data Augmentation:** TensorFlow provides a range of tools for preprocessing and data augmentation, including image and text preprocessing, data normalization, and more.

**Distributed Training:** TensorFlow supports distributed training across multiple devices, CPUs, and GPUs, allowing for faster training times and more efficient use of resources.

**Model Deployment:** TensorFlow allows for easy deployment of models to a variety of platforms, including mobile devices and the web.

**Visualization Tools:** TensorFlow provides a range of visualization tools foranalyzing model performance, including Tensor Board, which allows for real- time visualization of model training and performance.

### 5.3 PYCHARM

PyCharm is an integrated development environment (IDE) for Python programming language, developed by JetBrains. PyCharm provides features such as code completion, debugging, code analysis, refactoring, version control integration, and more to help developers write, test, and debug their Python codeefficiently. PyCharm is available in two editions: Community Edition (CE) and Professional Edition (PE). The Community Edition is a free, open-source versionof the IDE that provides basic functionality for Python development. The Professional Edition is a paid version of the IDE that provides advanced featuressuch as remote development, web development, scientific tools, database tools,

and more. PyCharm is available for Windows, macOS, and Linux operatingsystems. It supports Python versions 2.7, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, and 3.10.

**Features:**

- Intelligent code completion
- Syntax highlighting
- Code inspection
- Code navigation and search
- Debugging
- Testing
- Version control integration
- Web development support
- Scientific tools support
- Database tools support

**Integration with other JetBrains tools**

PyCharm's code completion feature can help speed up development by automatically suggesting code based on context and previously written code. It also includes a debugger that allows developers to step through code, set breakpoints, and inspect variables. PyCharm has integration with version controlsystems like Git, Mercurial, and Subversion. It also supports virtual environments, which allow developers to manage different Python installations and packages in isolated environments. The IDE also has features specifically geared towards web development, such as support for popular web frameworks like Django, Flask, and Pyramid. It includes tools for debugging, testing, and profiling web applications. PyCharm also provides scientific tools for data analysis, visualization, and scientific computing, such as support for NumPy, SciPy, and matplotlib. It includes tools for debugging, testing, and profiling web applications. It also includes tools for working with databases, such as PostgreSQL, MySQL, and Oracle. Overall, PyCharm is a powerful and feature-rich IDE that can greatly increase productivity for Python developers.

**Customization:**

PyCharm allows developers to customize the IDE to their liking. Users canchange the color scheme, fonts, and other settings to make the IDE morecomfortable to use. PyCharm

also supports plugins, which allow developers to extend the IDE with additional features.

**Collaboration:**

PyCharm makes it easy for developers to collaborate on projects. It supports integration with popular collaboration tools such as GitHub, Bitbucket,and GitLab. It also includes features for code reviews, task management, and team communication.

**Education:**

PyCharm provides a learning environment for Python programming language. PyCharm Edu is a free, open-source edition of PyCharm that includesinteractive courses and tutorials for learning Python. It provides an easy-to-use interface for beginners and includes features such as code highlighting, autocompletion, and error highlighting.

**Support:**

PyCharm has an active community of users who provide support through forums and social media. JetBrains also provides comprehensive documentation,tutorials, and training courses for PyCharm. For users who need more personalized support, JetBrains offers a paid support plan that includes email andphone support. The IDE also has features specifically geared towards web development, such as support for popular web frameworks like Django, Flask, and Pyramid. It includes tools for debugging, testing, and profiling web applications.

**Pricing**

PyCharm Community Edition is free and open-source. PyCharm Professional Edition requires a paid license, but offers a 30-day free trial.JetBrains also offers a subscription-based pricing model that includes access to all JetBrains IDEs and tools.

**Integrations:**

PyCharm integrates with a wide range of tools and technologies commonly used in Python development. It supports popular Python web frameworks like Flask, Django, Pyramid, and web2py. It also integrates with tools for scientific computing like NumPy, SciPy, and pandas. PyCharm also supports popular front- end technologies such as HTML, CSS, and JavaScript.

**Performance:**

PyCharm is known for its fast and reliable performance. It uses a combination of static analysis, incremental compilation, and intelligent caching to provide fast code completion and navigation. PyCharm also has a memory profiler that helps identify and optimize memory usage in Python applications.

**Ease of Use:**

PyCharm provides an intuitive and easy-to-use interface for developers. Ithas a well-organized menu structure, clear icons, and easy-to-navigate tabs. PyCharm also provides a variety of keyboard shortcuts and customizable keymaps that allow users to work efficiently without constantly switching between the mouse and keyboard.

**Community:**

PyCharm has a large and active community of developers who contribute to the development of the IDE. The PyCharm Community Edition is open-source, which means that anyone can contribute to its development. PyCharm also supports popular front- end technologies such as HTML, CSS, and JavaScript.

### 5.4 MY SQL

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi- user access to a number of databases. The MySQL development project has madeits source code available under the terms of the GNU General Public License, aswell as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

**Inter images**

MySQL is primarily an RDBMS and ships with no GUI tools to administerMySQL databases or manage data contained within the databases. Users may usethe included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database

structures, back up data, inspect status, and work with data records.

**Graphical**

The official MySQL Workbench is a free integrated environment developed by MySQL AB that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

# CHAPTER 6

# SYSTEM/SUBSYSTEM SPECIFICATION

## 6.1 MODULES

There are five modules implemented in our system

1. Framework Construction
2. Camera Capturing
3. Helmet Classification
4. Number Plate Recognition
5. Alert System

### 6.1.1 FRAMEWORK CONSTRUCTION

Intelligent traffic systems optimize the movement of automobiles over transport networks. This optimization includes Automatic Car License Plate Detection and Recognition. Research on Automatic car license plate detection LPR has gained momentum in recent years due to neural networks and deep learning. It can be applied to many areas like traffic law enforcement and road traffic monitoring. To identify License plate technologies such as computer vision and artificial intelligence algorithms can employed.

### 6.1.2 CAMERA CAPTURING

Since motorcycles are affordable and a daily mode of transport, there has been a rapid increase in motorcycle accidents due to the fact that most of the motorcyclists do not wear helmets which makes it an ever-present danger ever Implement binarization steps to separate the foreground from background scenes. Background subtraction is any technique which allows an image's foreground to be extracted for further processing (object recognition etc.). Manyapplications do not need to know everything about the evolution of movement ina video sequence, but only require the information of changes in the scene, because an image's regions of interest are objects (humans, cars, text etc.) in its foreground.

### 6.1.3 HELMET CLASSIFICATION

In this module implement object detection system using YOLO algorithm.Then detect the objects and draw bounding box on that object. Verify the featureswhich are contains the helmet objects. If helmet object not occurred means, forward to next module. For training our custom object detection model, we willneed a lot of images of objects which we're going to train nearly a few thousand. Number of images is directly proportional to accurate precision.

### 6.1.4 NUMBER PLATE RECOGNITION

In this module, implement number detection approach based on textstrokes values which is defined in the form of minimum and maximum values inorder to obtain the license plate only and remove other very small or very large identified objects which were outside the threshold range. The objects passed successfully through predefined threshold criterion were forwarded to the training process. In this module, text strokes in number plate detected using Conditional Random field. Detected texts are drawn as bounding box. Text can be recognized using Convolutional neural network algorithm. In this module implement CNN algorithm to recognize the detected text. Character segmentation is done on the binary image of the extracted license plate.

- Steps in CNN algorithms:
- Step 1: Randomly initialize the weights and biases.
- Step 2: feed the training sample.
- Step 3: Propagate the inputs forward; compute the net input and output ofeach unit in the hidden and output layers.
- Step 4: back propagate the error to the hidden layer.
- Step 5: update weights and biases to reflect the propagated errors.
- Step 6: terminating condition

### 6.1.5 ALERT SYSTEM

In this module, recognized user details are extracted from database which are extracted from trained database. CNNs will compare input images pixel by pixel or group of boxes. The regions that appearances for are called landscapes. By definition rough feature contests in roughly the similar places in two images,convolutional neural network

gets a lot of improved at sighted likeness than entire–image matching patterns. And send the fine amount details to appropriatethe user in the form of SMS alert.

## 6.2 SYSTEM DESCRIPTION

Helmet violation detection is a computer vision application that uses deeplearning algorithms to detect whether or not a person is wearing a helmet in a video frame. This technology has important applications in road safety, as helmetuse is a critical factor in reducing the risk of serious injury or death in motorcycleaccidents. The helmet violation detection system typically involves the collectionof a large dataset about helmets that are collected from Kaggle source. This dataset is used to train YOLO algorithm that can then accurately classify whethera person in a given video frame is wearing a helmet or not. The algorithm worksby identifying key features of the helmet, such as its shape, color, and position onthe person's head, and comparing these features to those of non-helmeted individuals. Once the algorithm has been trained, it can be deployed in a variety of settings, such as at traffic intersections or on the side of roads, to detect helmetviolations in real-time. By helping to ensure that motorcycle riders are wearing helmets, this technology can help to prevent serious injuries and save lives on theroads. And also recognize the number plate to provide the fine amount to users. The helmet violation detection system typically involves the collectionof a large dataset about helmets that are collected from Kaggle source. This dataset is used to train YOLO algorithm that can then accurately classify whethera person in a given video frame is wearing a helmet or not.

### 6.2.1 YOLO ALGORITHM

- YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals.
- YOLO algorithm works using the following three techniques:
- Residual blocks
- A bounding box is an outline that highlights an object in an image.
- Every bounding box in the image consists of the following attributes:
- Width (bw)
- Height (bh)
- Class (for example, person, car, traffic light, etc.)- This is represented bythe letter c.

- Bounding box center (bx,by)
- Bounding box regression
- Bounding box regression
- A bounding box is an outline that highlights an object in an image.
- Every bounding box in the image consists of the following attributes:
- Width (bw)
- Height (bh)
- Class (for example, person, car, traffic light, etc.)- This is represented bythe letter c.
- Bounding box center (bx,by)
- Intersection Over Union (IOU)
- Intersection over union (IOU) is a phenomenon in object detection that describes how boxes overlap. YOLO uses IOU to provide an output box that surrounds the objects perfectly.

### 6.2.2 CONVOLUTIONAL NEURAL NETWORK ALGORITHM

Artificial Neural Networks (ANN) are capable of learning and may thus be trained to recognize patterns, develop solutions, predict future occurrences, and classify data. The use of ANN for traffic-related activities is well established. The way its separate computing parts are coupled, as well as the intensities of these connections or weights, determine how neural networks learn and behave.

- **Data Collection:** Collect a large dataset of images of number plates.
- **Data Pre-Processing:** Pre-process the images by applying image enhancement techniques to improve image quality, such as contrast stretching and gamma correction.
- **Character Segmentation:** Segment the individual characters on the numberplate by identifying the contours of each character and cropping the imagesto isolate each character.
- **Data Labeling:** Label each character image with its corresponding alphanumeric value.
- **CNN Model Building:** Build a CNN model with multiple layers of convolutional and pooling layers that can learn to identify the unique features of each character.

- **Model Training:** Train the CNN model using the labeled character images by adjusting the weights of the neurons to minimize the error between thepredicted and actual alphanumeric values.
- **OCR Algorithm:** Build an OCR algorithm that can recognize the charactersin the segmented images by combining the predictions of the CNN model.
- **Number Plate Recognition:** Apply the OCR algorithm to the entire numberplate image to recognize the characters and obtain the alphanumeric valueof the number plate.

# CHAPTER 7
# SYSTEM DESIGN

## 7.1 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determinehow the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

### DFD LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from anexternal agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order forthe system to do its job, and shows the flow of data between the various parts of the system.

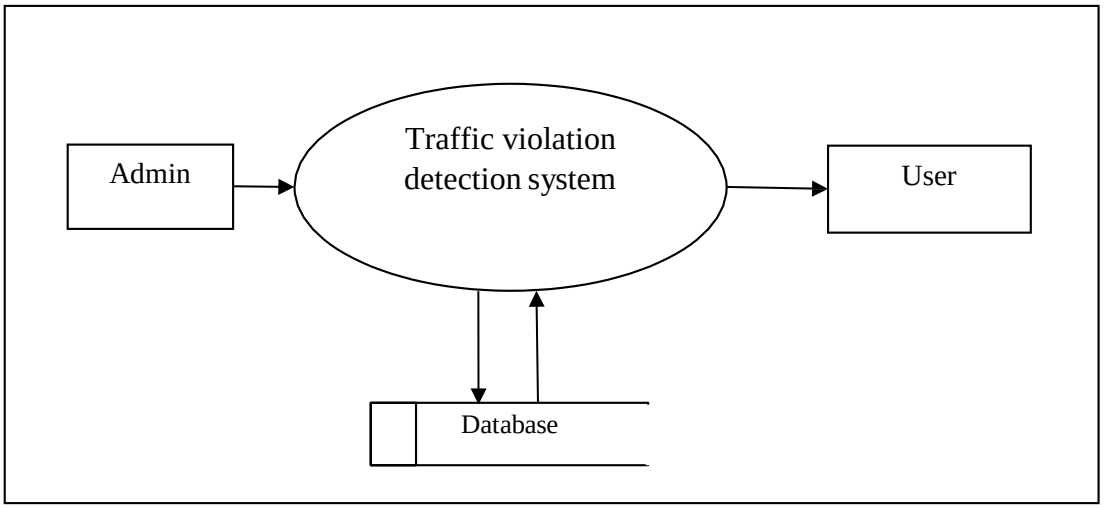


**Fig.7.1. DFD Level 0**

## DFD LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlightsthe main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageableon screen or paper.

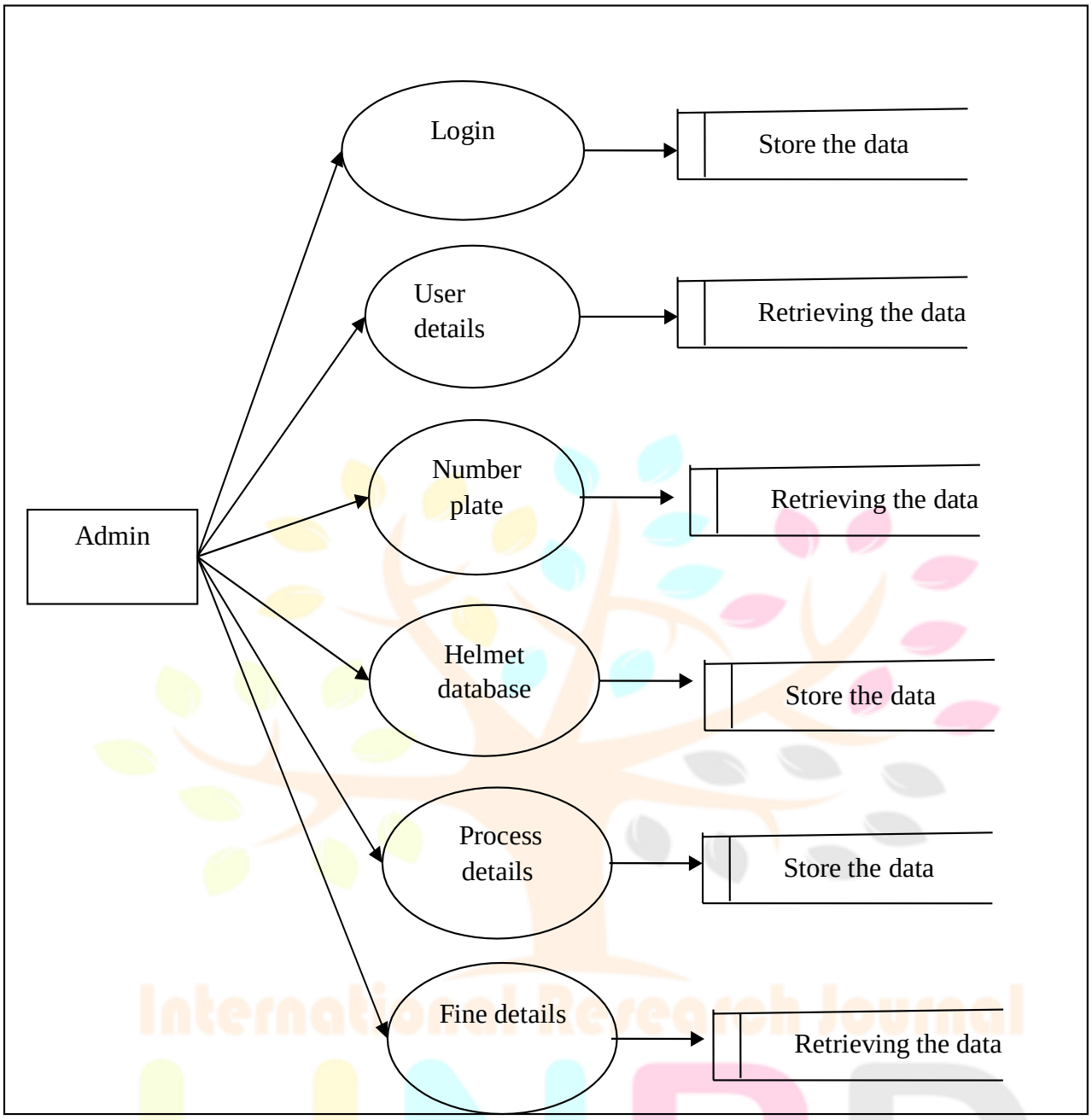


**Fig.7.2. DFD Level 1**

## DFD LEVEL 2

A Data Flow Diagram (DFD) tracks processes and their data paths within the business or system boundary under investigation. A DFD defines each domain boundary and illustrates the logical movement and transformation of data withinthe defined boundary. The diagram shows 'what' input data enters the domain, 'what' logical processes the domain applies to that data, and 'what' output data leaves the domain. Essentially, a DFD is a tool for process modeling and one of the oldest

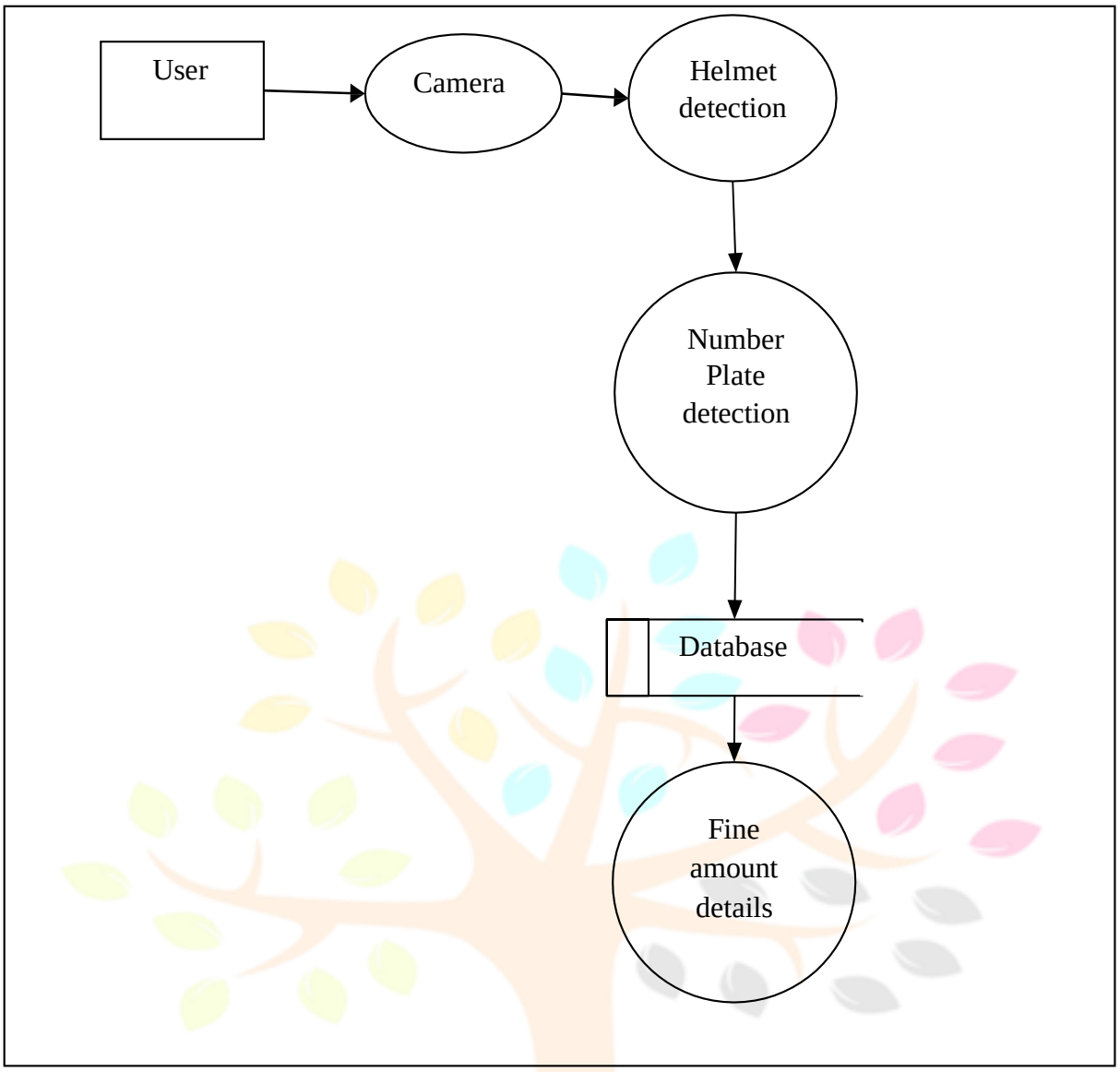


**Fig.7.3. DFD Level 2**

## 7.2 USECASE DIAGRAM

A use case diagram is a type of UML diagram that represents the interactions between an actor (a user or system) and a system under various scenarios. The diagram provides a visual representation of the system's functionalities and the interactions between the actors and the system. The greatest strategy to increase predictive power and the ability to generalise across several new datasets is to employ larger training datasets. The back propagation algorithm can be used to classify the disease.

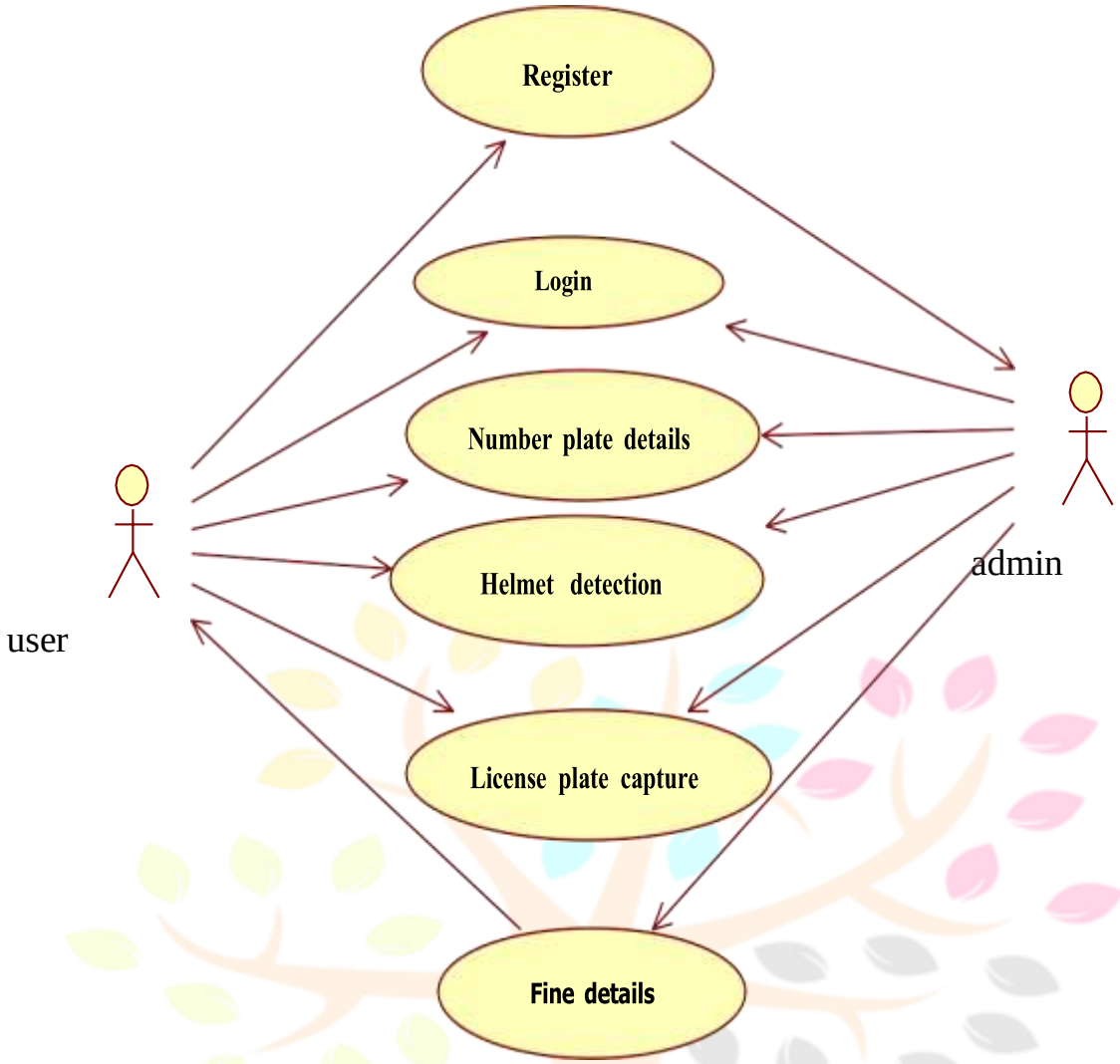


**Fig.7.4. Use Case Diagram**

## 7.3 CLASS DIAGRAM

A class diagram is a type of visual representation used in software development to depict the classes, attributes, and methods of a system, as well asthe relationships that exist between them. In this type of diagram, each class is represented as a box that includes its name, along with its attributes and methods.The attributes of a class are represented as variables that describe the characteristics of the class, while methods are represented as functions that define the behavior of the class.

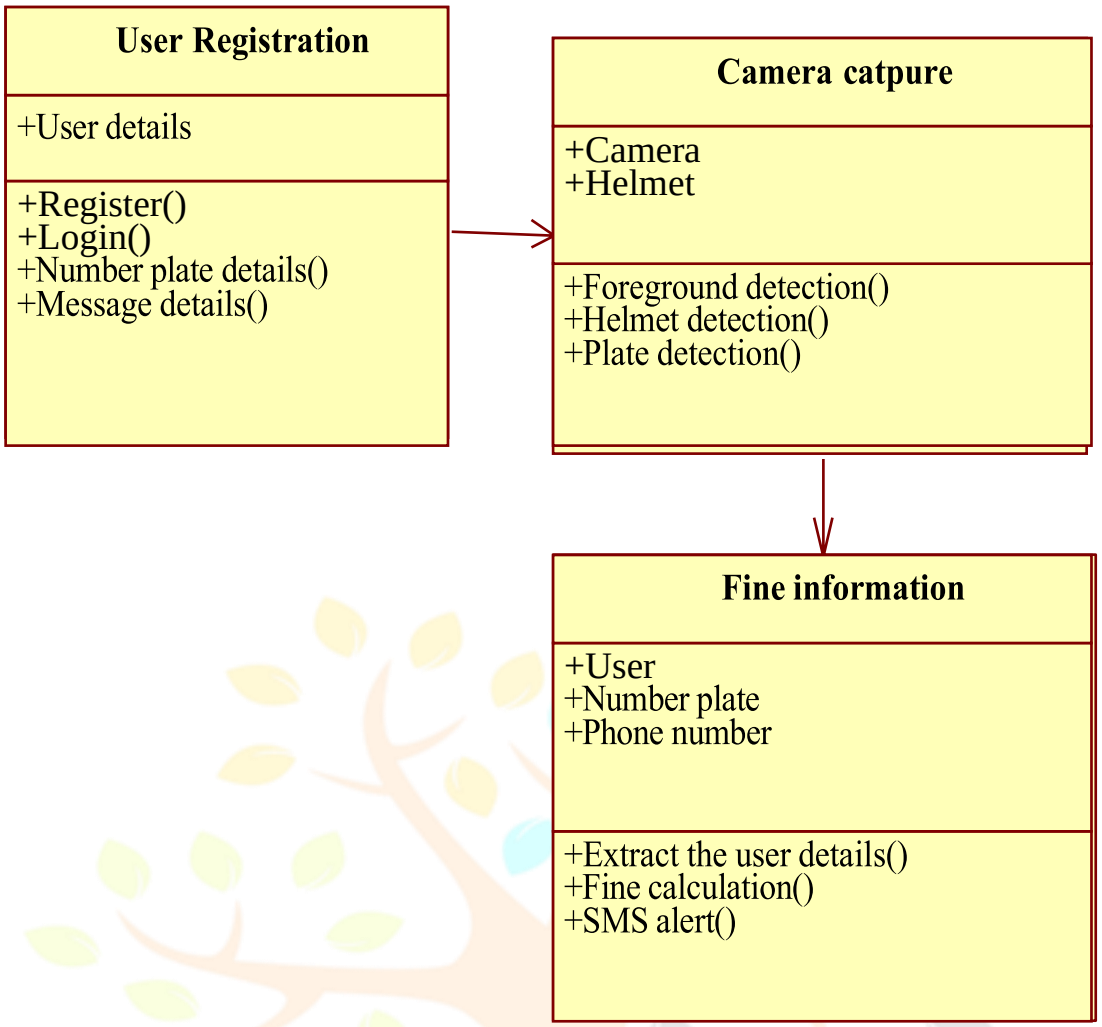


**Fig.7.5. Class Diagram**

## 7.4 ACTIVITY DIAGRAM

Activity diagrams show the workflow from a start point to the finish pointdetailing the many decision paths that exist in the progression of events containedin the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities.

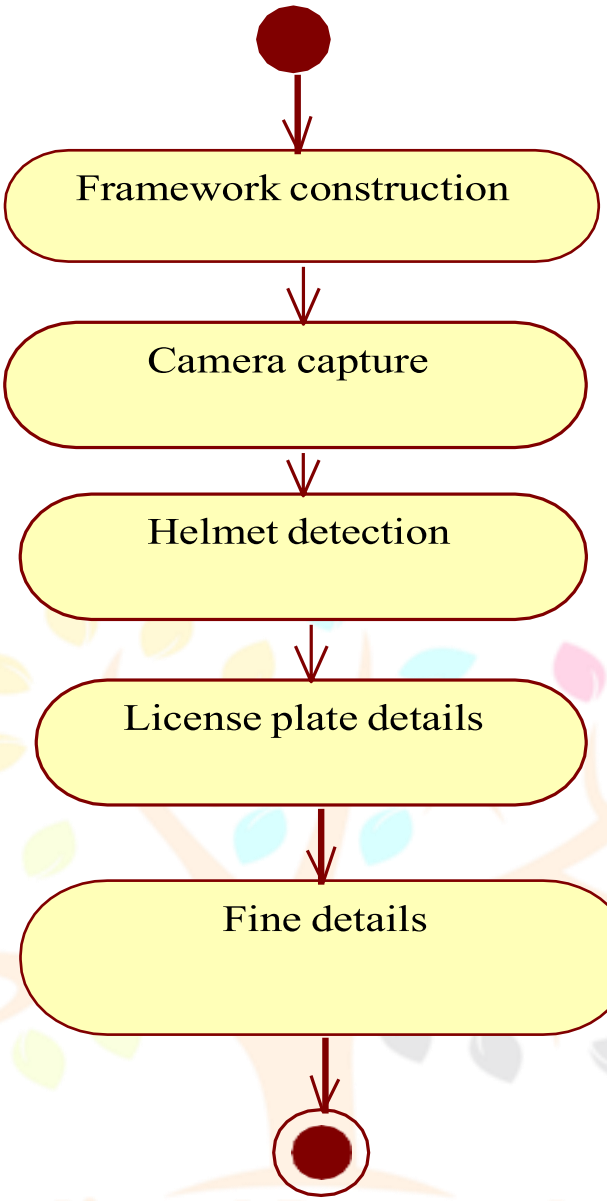


**Fig.7.6. Activity Diagram**

## 7.5 SEQUENCE DIAGRAM

A sequence diagram is a type of UML (Unified Modeling Language) diagram that illustrates the interactions between objects in a system in a sequential order. It shows the sequence of messages exchanged between objects over time, and can be used to represent a variety of scenarios, such as the flow of control between objects, the interactions between different components of a system, and the sequence of events that occur during a particular process. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

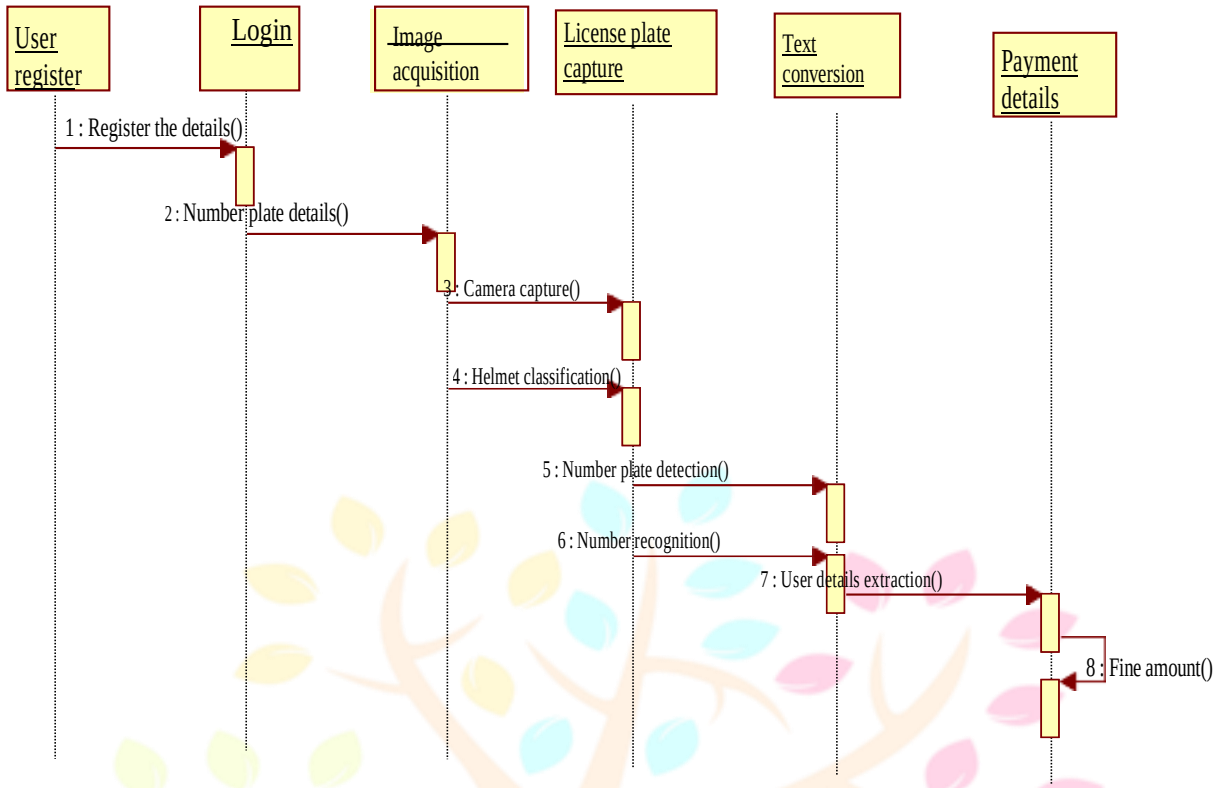


**Fig.7.7. Sequence Diagram**

# CHAPTER 8

# SOFTWARE TESTING

## 8.1 TYPES OF TESTING

Software testing is a method of assessing the functionality of a software program. There are many different types of software testing but the two main categories are dynamic testing and static testing. Dynamic testing is an assessment that is conducted while the program is executed; static testing, on theother hand, is an examination of the program's code and associated documentation. Dynamic and static methods are often used together.

1. Unit Test
2. System Test
3. Integrating Test
4. Functional Test
5. Black Box Test
6. White Box Test

### 8.1.1 UNIT TESTING

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the projectis developed.

### 8.1.2 INTEGRATION TESTING

Integration testing is a type of software testing that focuses on verifying that the individual components of a software system work together as expected. The objective of integration testing is to ensure that the software system as a whole function correctly, and that the individual components interact with each other as intended.

### 8.1.3 FUNCTIONAL TESTING

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifyingthat the module performs its intended functions as stated in the specification andestablishing confidence that a program does what it is supposed to do.

### 8.1.4 SYSTEM TESTING

Testing is a set activity that can be planned and conducted systematically.Testing begins at the module level and works towards the integration of the entirecomputers-based system. Nothing is complete without testing, as it is a vital success of the system.

### 8.1.5 BLACK BOX TESTING

Testing without knowledge of the internal workings of the item being tested. Tests are usually functional. This testing can be done by the user who hasno knowledge of how the shortest path is found.

### 8.1.6 WHITE BOX TESTING

Testing based on an analysis of internal workings and structure of a piece of software. This testing can be done using the percentage value of load and energy. The tester should know what exactly is done in the internal program. Includes techniques such as Branch Testing and Path Testing. Also known as Structural Testing and Glass Box Testing.

## 8.2 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Acceptance testing is typically performed after the individual components of the software system have been tested and verified to work correctly.

# CHAPTER 9

# SYSTEM IMPLEMENTATION

## 9.1 USER TRAINING

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of thesystems. The people are not sure that the software is meant to make their job easier.

- The active user must be aware of the benefits of using the systems.
- Their confidence in the software built up.
- Proper guidance is impaired to the user so that he is comfortable in using the application.

Before going ahead and viewing the systems, the user must know that for viewing the results, the server program should be running in the server. If the server objectis not running on the server, the actual processes will not take place.

## 9.2 TRAINING ON THE APPLICATION SOFTWARE

To achieve the objective and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important.

Education is complementary to training. It brings life to formal training by explaining the back ground to the resources for them. education involves creating the right atmosphere and motivating user staff. education information can make training more interesting and more understandable. Such maintenance may also include minor parts replacement that does not require the person performing the work to have highly technical skills or to perform internal alignment. Useful tips and guidance is given inside the application itself to the user.

## 9.3 OPERATIONAL DOCUMENTATION

After providing the necessary basic training on the computer awareness, the userswill have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the dataentered. This training may be different across different user groups and across different levels of hierarchy.

## 9.4 SYSTEM MAINTENANCE

Once the implementation plan is decided, it is essential that the user of thesystem is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips andguidance is given inside the application itself to the user. The system is developeduser friendly so that the user can work the system from the tips given in the application itself. The results obtained from the evaluation process help the organization to determine whether its information systems are effective and efficient or otherwise. the process of monitoring, evaluating, and modifying of existing information systems to make required or desirable improvements may betermed as system maintenance.

## 9.5 CORRECTIVE MAINTENANCE

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is the important aspect in the software development life cycle.

The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environment changes, which affect a system which is being implemented.

## 9.6 ADAPTIVE MAINTENANCE

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer the process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

## 9.7 PERCEPTIVE MAINTENANCE

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore, adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

Perceptive Process Design is a user-friendly business process modelling environment that allows business users to diagram, model, display, and documentand publishes business process maps and meets process-specific compliance requirements. Perceptive Process Enterprise is a case-based business process management toolthat supports complex case-handling and work process execution and automation for a variety of industries and organizations.

## 9.8 PREVENTIVE MAINTENANCE

The third activity that may be applied to a definition of maintenance occurs whena software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance. It is a regular and routine action taken on equipment in order to prevent its breakdown.

# CHAPTER 10

# CONCLUSION AND FUTURE ENHANCEMENT

## 10.1 CONCLUSION

In this project we have described a framework for automatic detection of motorcycle riders without helmet from real time camera capturing and automaticretrieval of vehicle license number plate for such motorcyclists. The use of Convolutional Neural Networks (CNN) and transfer learning will help in achieving good accuracy for detection of motorcyclists not wearing helmets. But,only detection of such motorcyclists is not sufficient for taking action against them. So, the system will also recognize the number plates of their motorcycles and store them. In this project, we have used the YOLO v3 for identification of real time person with and without helmets. YOLO is suitable to detect the singleobject from the image, YOLO has a limitation that if there are multiple objects

ina single cell then YOLO is not suitable to all objects. And also accomplished deep learning based automatic license plate recognition model for Indian road users. Results denote that the preferred technique perform better than the existingmethods by far in energizing datasets of Indian fonts with high irregularities, containing Number plates and successfully created a custom dataset of Indian font variants Successfully trained the model with Sequential CNN algorithm. Thestored number plates can be then used by Transport Office to get information about the motorcyclists from their database of licensed vehicles. Concerned motorcyclists can then be penalized.

## 10.2 FUTURE ENHANCEMENTS

In future, we can extend the framework analyze various types of traffic violations and with embedded with hardware system. We can also predict the traffic misbehaving in various types' vehicles and also use other deep learning algorithms to improve the accuracy. the system will also recognize the number plates of their motorcycles and store them.

## APPENDIX-1

## SOURCE CODE

```
from flask import Flask, render_template, flash, request, sessionfrom
flask import render_template, redirect, url_for, request import
mysql.connector
import datetimeimport
time
app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
def homepage():
    return render_template('index.html')
@app.route("/Home")
def Home():
    return render_template('index.html')
@app.route("/AdminLogin")
```

```
def AdminLogin():
    return render_template('AdminLogin.html')
@app.route("/UserLogin")
def UserLogin():
    return render_template('UserLogin.html')
@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')
@app.route("/adminlogin", methods=['GET', 'POST'])def
adminlogin():
    error = None
    if request.method == 'POST':
      if request.form['uname'] == 'admin' or request.form['password'] == 'admin':conn =
          mysql.connector.connect(user='root', password='',
host='localhost', database='1numberhelmetdb')cursor =
          conn.cursor()
          cur = conn.cursor() cur.execute("SELECT *
          FROM regtb")data = cur.fetchall()
          return render_template('AdminHome.html', data=data)else:
       return render_template('index.html', error=error)
@app.route("/AdminHome")
def AdminHome():
    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1numberhelmetdb')
    cursor = conn.cursor()cur =
    conn.cursor()
    cur.execute("SELECT * FROM regtb")data =
    cur.fetchall()
    return render_template('AdminHome.html', data=data)
@app.route("/remove")
def remove():
    did =  request.args.get('did')
```

```
    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1numberhelmetdb')

    cursor = conn.cursor()

    cursor.execute("delete from regtb  where Id='"+ did +"' ")
    conn.commit()

    conn.close()

    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1numberhelmetdb')

    # cursor = conn.cursor()cur =

    conn.cursor()

    cur.execute("SELECT * FROM regtb ")data =

    cur.fetchall()

    return render_template('AdminHome.html', data=data )

@app.route("/Report")

def Report():

    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1numberhelmetdb')

    cur = conn.cursor() cur.execute("SELECT *

    FROM entrytb")data = cur.fetchall()

    return render_template('AdminReport.html', data=data)import

cv2

import numpy as np

from skimage.filters import threshold_localimport

tensorflow as tf

from skimage import measureimport

imutils

import pytesseractimport

re

import mysql.connector

def sort_cont(character_contours):"""

    def__init__(self):

        self.min_area = 4500 # minimum area of the plate

        self.max_area = 30000  # maximum area of the plate
```

```
 self.element_structure = cv2.getStructuringElement(shape=cv2.MORPH_RECT,
ksize=(22, 3))

    def preprocess(self, input_img):

        imgBlurred = cv2.GaussianBlur(input_img, (7, 7), 0)  # old window was
(5,5)

gray
gray = cv2.cvtColor(imgBlurred, cv2.COLOR_BGR2GRAY)  # convert tosobelx =
cv2.Sobel(gray, cv2.CV_8U, 1, 0, ksize=3)  # sobelX to get the
vertical edges

        ret2, threshold_img = cv2.threshold(sobelx, 0, 255,cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

 element = self.element_structure morph_n_thresholded_img =

        threshold_img.copy()

        cv2.morphologyEx(src=threshold_img, op=cv2.MORPH_CLOSE,kernel=element,
dst=morph_n_thresholded_img)

        return morph_n_thresholded_im

 def extract_contours(self, after_preprocess):

        _, contours, _ = cv2.findContours(after_preprocess,mode=cv2.RETR_EXTERNAL,

       method=cv2.CHAIN_APPROX_NONE) return

        contours

    def clean_plate(self, plate):

        gray = cv2.cvtColor(plate, cv2.COLOR_BGR2GRAY)thresh =

        cv2.adaptiveThreshold(gray, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)

        _, contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)

        if contours:

            areas = [cv2.contourArea(c) for c in contours]

            max_index = np.argmax(areas)  # index of the largest contour in the area array

max_cnt = contours[max_index]


ax_cntArea = areas[max_index]

x, y, w, h = cv2.boundingRect(max_cnt)rect =

cv2.minAreaRect(max_cnt) rotatedPlate = plate

if not self.ratioCheck(max_cntArea, rotatedPlate.shape[1],
rotatedPlate.shape[0]):
```

```
            return plate, False, None
        return rotatedPlate, True, [x, y, w, h]else:
        return plate, False, None
    def check_plate(self, input_img, contour):
        min_rect = cv2.minAreaRect(contour) if
        self.validateRatio(min_rect):
            x, y, w, h = cv2.boundingRect(contour)
            after_validation_img = input_img[y:y + h, x:x + w]after_clean_plate_img,
            plateFound, coordinates =
self.clean_plate(after_validation_img)if
            plateFound:
                characters_on_plate =
self.find_characters_on_plate(after_clean_plate_img)
                if (characters_on_plate is not None and len(characters_on_plate) ==10):
x1, y1, w1, h1 = coordinatescoordinates = x1 + x, y1 + y
                    after_check_plate_img = after_clean_plate_img
                    return after_check_plate_img, characters_on_plate, coordinatesreturn None,
        None, None
    def find_possible_plates(self, input_img):"""
        Finding all possible contours that can be plates"""
        plates = [] self.char_on_plate = []
        self.corresponding_area = []
 self.after_preprocess = self.preprocess(input_img) possible_plate_contours =
        self.extract_contours(self.after_preprocess)
 for cnts in possible_plate_contours:
            plate, characters_on_plate, coordinates = self.check_plate(input_img,cnts)
if plate is not None: plates.append(plate)
    self.char_on_plate.append(characters_on_plate)
    self.corresponding_area.append(coordinates)
if (len(plates) > 0):
            return plateselse:
            return None
```

```
def find_characters_on_plate(self, plate):
    charactersFound = segment_chars(plate, 400)if
    charactersFound:
        return charactersFound#
PLATE FEATURES
def ratioCheck(self, area, width, height):min =
    self.min_area
    max = self.max_area
    ratioMin = 3
    ratioMax = 6
    ratio = float(width) / float(height)if ratio <
    1:
        ratio = 1 / ratio
    if (area < min or area > max) or (ratio < ratioMin or ratio > ratioMax):return False
    return True
def preRatioCheck(self, area, width, height):min =
    self.min_area
    max = self.max_area
    ratioMin = 2.5
    ratioMax = 7
    ratio = float(width) / float(height)if ratio
    < 1:
        ratio = 1 / ratio
    if (area < min or area > max) or (ratio < ratioMin or ratio > ratioMax):return False
    return True
def validateRatio(self, rect):
    (x, y), (width, height), rect_angle = rectif (width
    > height):
        angle = -rect_angleelse:
method=cv2.CHAIN_APPROX_NONE)
    return contours
```

```
    def clean_plate(self, plate):
        gray = cv2.cvtColor(plate, cv2.COLOR_BGR2GRAY)
        thresh = cv2.adaptiveThreshold(gray, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)
        _, contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
    if contours:
            areas = [cv2.contourArea(c) for c in contours]
            max_index = np.argmax(areas)  # index of the largest contour in the area
array
    max_cnt = contours[max_index] max_cntArea =
            areas[max_index]
            x, y, w, h = cv2.boundingRect(max_cnt)rect =
            cv2.minAreaRect(max_cnt) rotatedPlate = plate
            if not self.ratioCheck(max_cntArea, rotatedPlate.shape[1],rotatedPlate.shape[0]):
                return plate, False, None
            return rotatedPlate, True, [x, y, w, h]else:
            return plate, False, None
    def check_plate(self, input_img, contour):
        min_rect = cv2.minAreaRect(contour) if
        self.validateRatio(min_rect):
            x, y, w, h = cv2.boundingRect(contour) after_validation_img =
            input_img[y:y + h, x:x + w]
            after_clean_plate_img, plateFound, coordinates =
self.clean_plate(after_validation_img)
            if plateFound:
                characters_on_plate =
self.find_characters_on_plate(after_clean_plate_img)
                    if (characters_on_plate is not None and len(characters_on_plate) ==
10):
            x1, y1, w1, h1 = coordinatescoordinates = x1 + x, y1
            + y
            after_check_plate_img = after_clean_plate_img
```

```
        return after_check_plate_img, characters_on_plate, coordinatesreturn None, None,
None
    def find_possible_plates(self, input_img):"""
      Finding all possible contours that can be plates"""
      plates = [] self.char_on_plate = []
      self.corresponding_area = []
self.after_preprocess = self.preprocess(input_img)
      possible_plate_contours = self.extract_contours(self.after_preprocess)for cnts in
possible_plate_contours:
          plate, characters_on_plate, coordinates = self.check_plate(input_img,
cnts)
  if plate is not None:
     plates.append(plate)
     self.char_on_plate.append(characters_on_plate)
     self.corresponding_area.append(coordinates)
if (len(plates) > 0):
          return plateselse:
          return None
def find_characters_on_plate(self, plate): charactersFound
      = segment_chars(plate, 400) if charactersFound:
          return charactersFound#
PLATE FEATURES
   def ratioCheck(self, area, width, height):min =
      self.min_area
      max = self.max_area
        ratioMin = 3
      ratioMax = 6
  ratio = float(width) / float(height)if ratio
      < 1:
          ratio = 1 / ratio
if (area < min or area > max) or (ratio < ratioMin or ratio > ratioMax):return
          False
```

```
        return True
def preRatioCheck(self, area, width, height):min =
        self.min_area
        max = self.max_area
          ratioMin = 2.5
        ratioMax = 7
  ratio = float(width) / float(height)if ratio
        < 1:
            ratio = 1 / ratio
if (area < min or area > max) or (ratio < ratioMin or ratio > ratioMax):return
            False
        return True
def validateRatio(self, rect):
        (x, y), (width, height), rect_angle = rectif (width
> height):
            angle = -rect_angleelse:
            angle = 90 + rect_angleif angle
 > 15:
            return False
        if (height == 0 or width == 0):return
            False
           area = width * height
        if not self.preRatioCheck(area, width, height):return False
        else:
            return True class
NeuralNetwork:
    def__init__(self):
        self.model_file = "./model/binary_128_0.50_ver3.pb" self.label_file
        = "./model/binary_128_0.50_labels_ver2.txt"self.label =
        self.load_label(self.label_file)
        self.graph = self.load_graph(self.model_file)self.sess =
        tf.Session(graph=self.graph)
```

```
def load_graph(self, modelFile):graph =
    tf.Graph()

    graph_def = tf.GraphDef()
    with open(modelFile, "rb") as f:
        graph_def.ParseFromString(f.read())
    with graph.as_default():
        tf.import_graph_def(graph_def)
    return graph
def load_label(self, labelFile):label =
    []
    proto_as_ascii_lines = tf.gfile.GFile(labelFile).readlines()for l in
    proto_as_ascii_lines:
        label.append(l.rstrip())return
    label
def convert_tensor(self, image, imageSizeOuput):"""
    takes an image and tranform it in tensor"""
    image = cv2.resize(image, dsize=(imageSizeOuput, imageSizeOuput),
interpolation=cv2.INTER_CUBIC)
    np_image_data = np.asarray(image)
    np_image_data = cv2.normalize(np_image_data.astype('float'), None, -0.5,
.5, cv2.NORM_MINMAX)
    np_final = np.expand_dims(np_image_data, axis=0)return
    np_final
    def label_image(self, tensor):
input_name = "import/input"
    output_name = "import/final_result"
input_operation = self.graph.get_operation_by_name(input_name) output_operation =
    self.graph.get_operation_by_name(output_name)
```

```
        break
weightsPath = os.path.abspath("./yolov3-helmet.weights")
configPath = os.path.abspath("./yolov3-helmet.cfg")
# print(configPath, "\n", weightsPath)
flagg1 = 0
flagg2 = 0
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]
vs = cv2.VideoCapture(0)
writer = None
(W, H) = (None, None)
flag = True
face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
while True:
    # read the next frame from the file
    (grabbed, frame) = vs.read()

    # if the frame was not grabbed, then we have reached the end
    # of the stream
    if not grabbed:
        break;
  # if the frame dimensions are empty, grab them
    if W is None or H is None:
        (H, W) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),
                                 swapRB=True, crop=False)
    net.setInput(blob)
    start = time.time()
    layerOutputs = net.forward(ln)
    end = time.time()
```

```
# initialize our lists of detected bounding boxes, confidences,
# and class IDs, respectively
boxes = []
confidences = []
classIDs = []
# loop over each of the layer outputs
for output in layerOutputs:
    # loop over each of the detections
    for detection in output:
        # extract the class ID and confidence (i.e., probability)
        # of the current object detection
        scores = detection[5:]
        classID = np.argmax(scores)
        confidence = scores[classID]
        # filter out weak predictions by ensuring the detected
        # probability is greater than the minimum probability
        if confidence > args["confidence"]:
            # scale the bounding box coordinates back relative to
            # the size of the image, keeping in mind that YOLO
            # actually returns the center (x, y)-coordinates of
            # the bounding box followed by the boxes' width and
            # height
            box = detection[0:4] * np.array([W, H, W, H])
            (centerX, centerY, width, height) = box.astype("int")
            # use the center (x, y)-coordinates to derive the top
            # and and left corner of the bounding box
            x = int(centerX - (width / 2))
            y = int(centerY - (height / 2))
            # update our list of bounding box coordinates,
            # confidences, and class IDs
            boxes.append([x, y, int(width), int(height)])
```

```
            confidences.append(float(confidence))
            classIDs.append(classID)
# apply non-maxima suppression to suppress weak, overlapping
# bounding boxes
idxs = cv2.dnn.NMSBoxes(boxes, confidences, args["confidence"],
                        args["threshold"])
# ensure at least one detection exists
if len(idxs) > 0:
    # loop over the indexes we are keeping
    for i in idxs.flatten():
        # extract the bounding box coordinates
        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])
        if (LABELS[classIDs[i]] in final_classes):
            # playsound('alarm.wav')
            if (flag):
                # alert()
                flag = False
                # async_email(LABELS[classIDs[i]])
            color = [int(c) for c in COLORS[classIDs[i]]]
            cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
            text = "{}: {:.4f}".format(LABELS[classIDs[i]],
                                       confidences[i])
            cv2.putText(frame, text, (x, y - 5),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
            cv2.imshow("Color", frame)
            flagg1 += 1
            # print(flag)
            if (flagg1 == 10):
                print("Helmet Found!")
```

```
    else:
        flag = True
        print(0)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5)
        for x, y, w, h in faces:
            frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)
        cv2.imshow('Color', frame)
flagg2 += 1
        # print(flag)
 if (flagg2 == 10):
            flagg2 = 0
            print("Helmet Not  Found!")
    if cv2.waitKey(1) == ord('q'):
        break
# release the webcam and destroy all active windows
vs.release()
```

# APPENDIX-2

## SCREENSHOTS

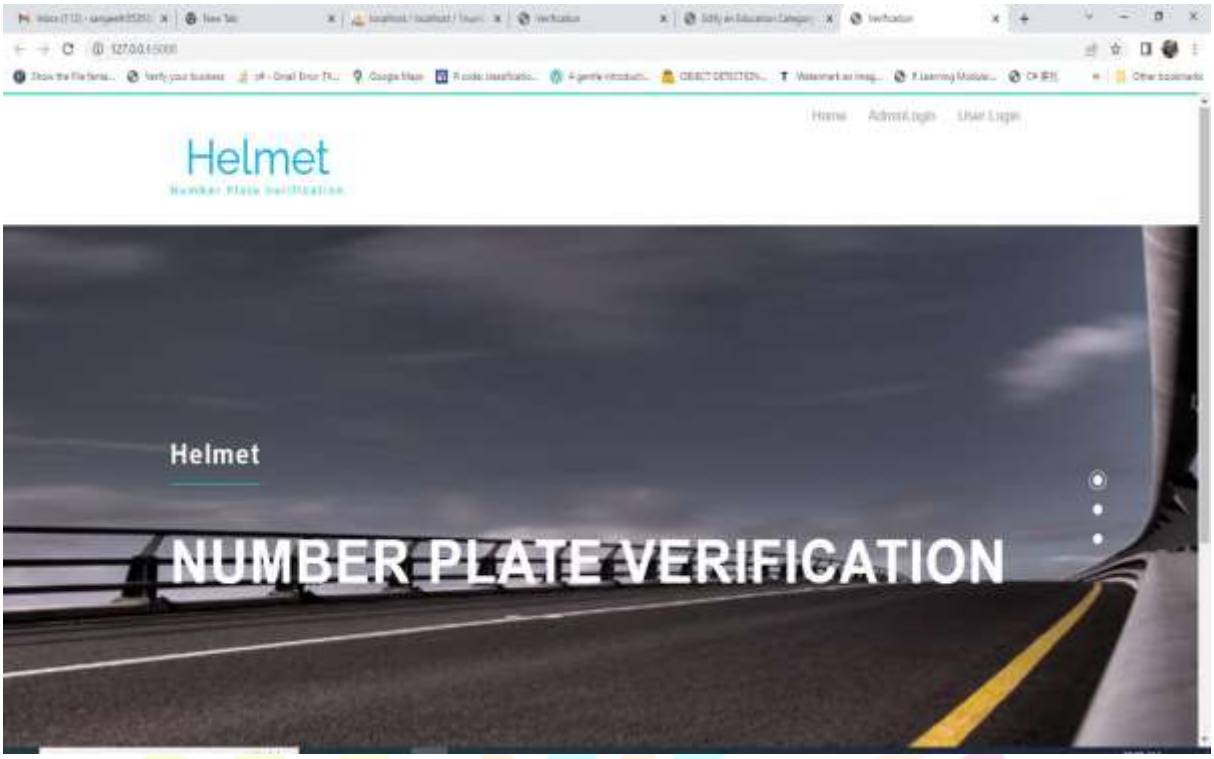


**Fig.1. Home Page**

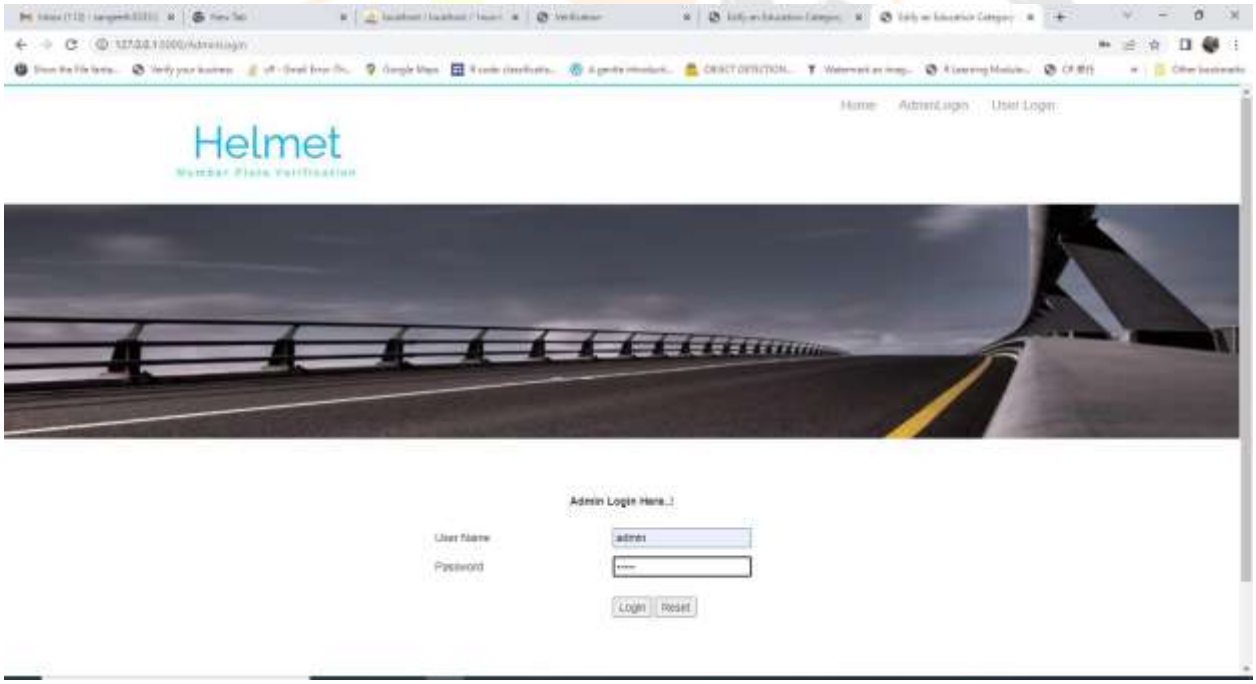


**Fig.2. Admin Login**

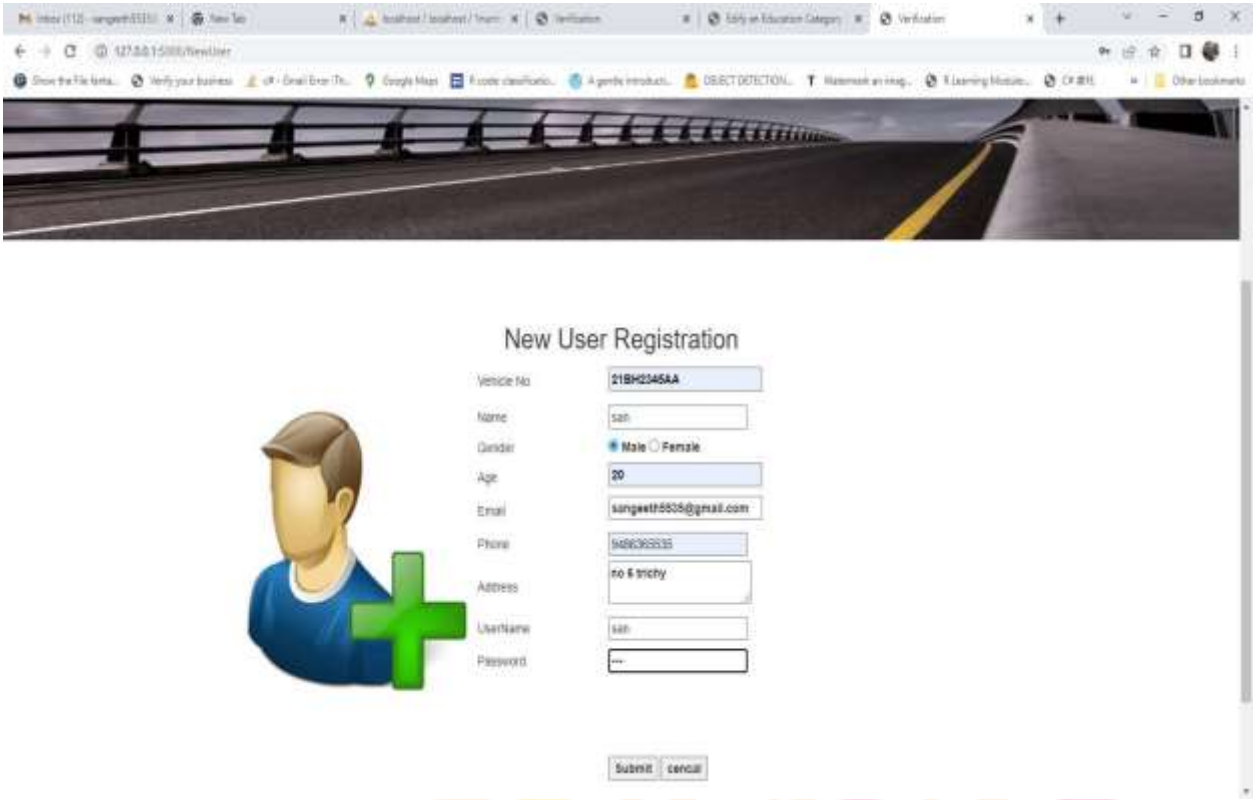


**Fig.3. New User Registration**

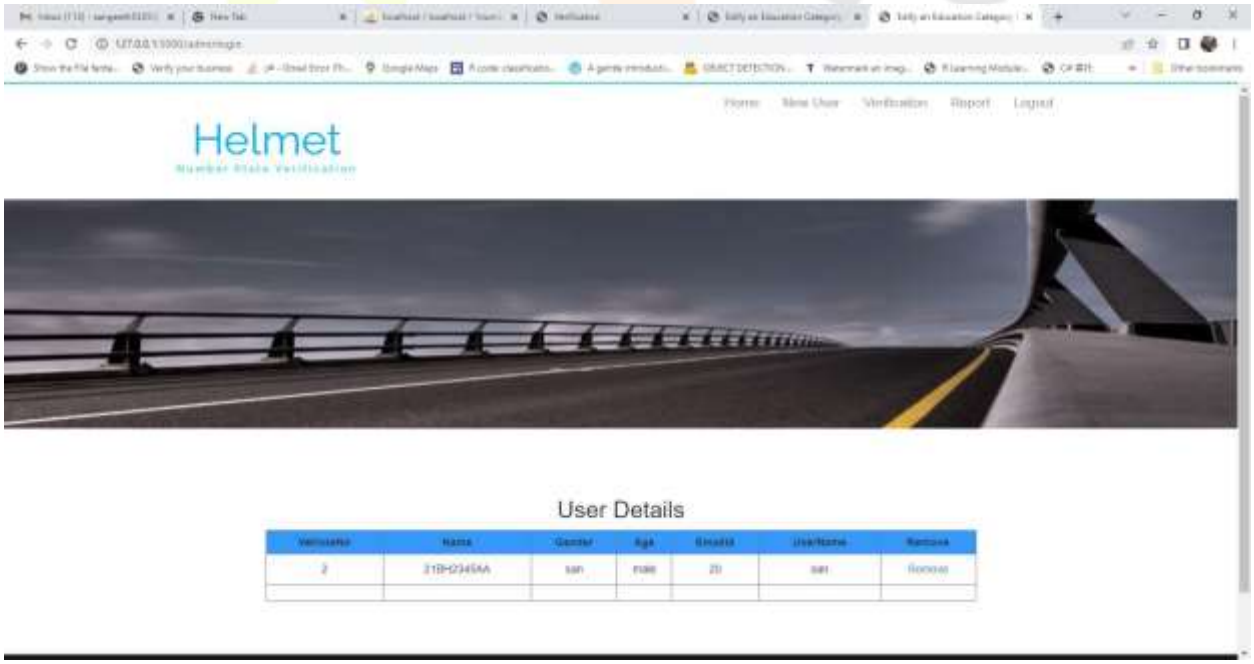


**Fig.4. User Details**

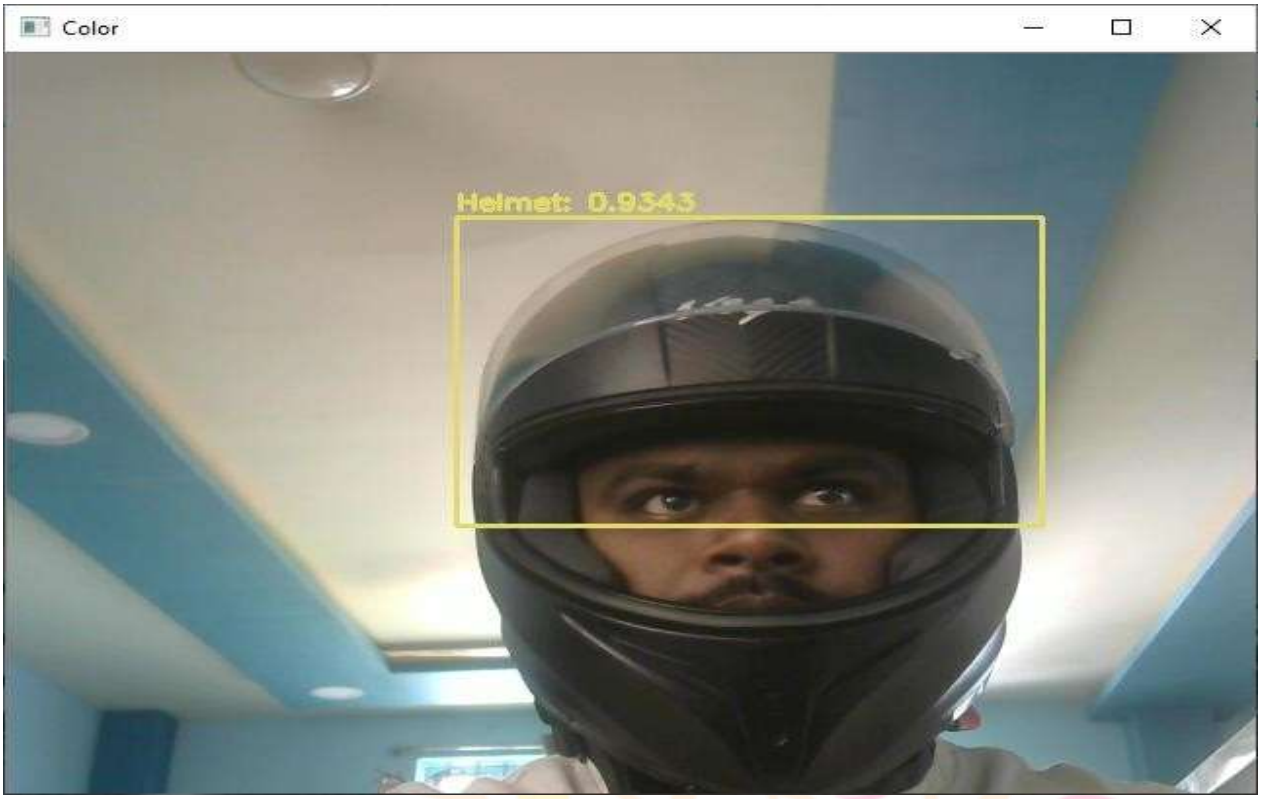


**Fig.5. Helmet Detection**

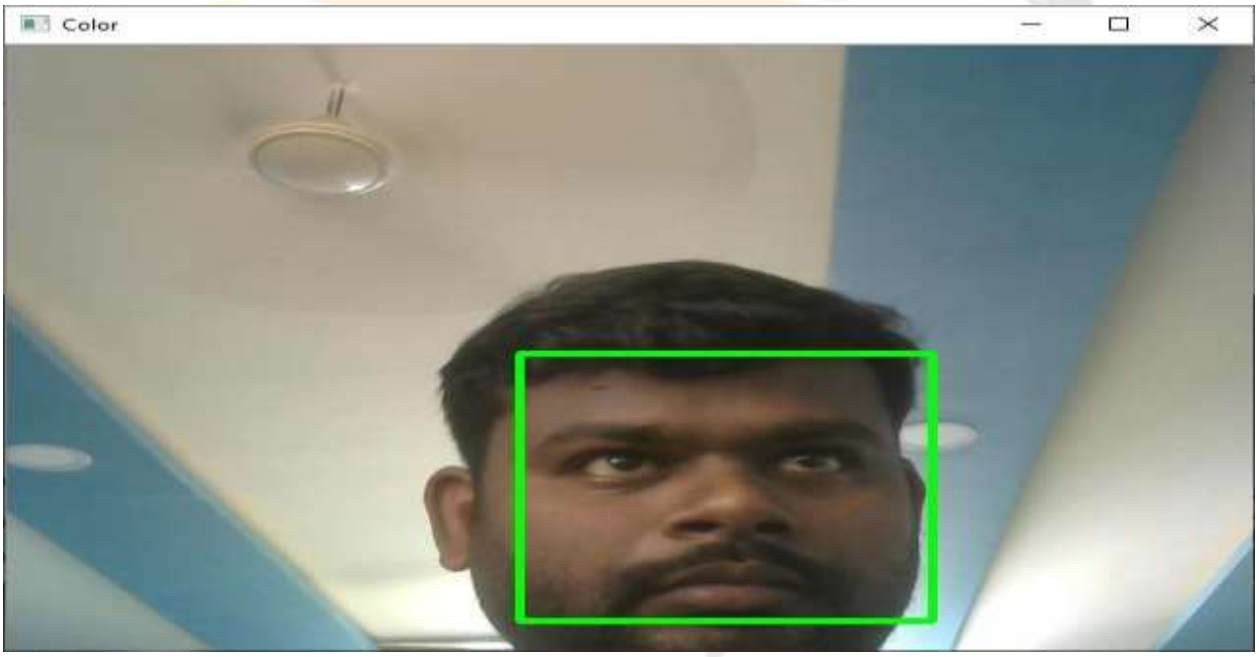


**Fig.6. No Helmet Detection**

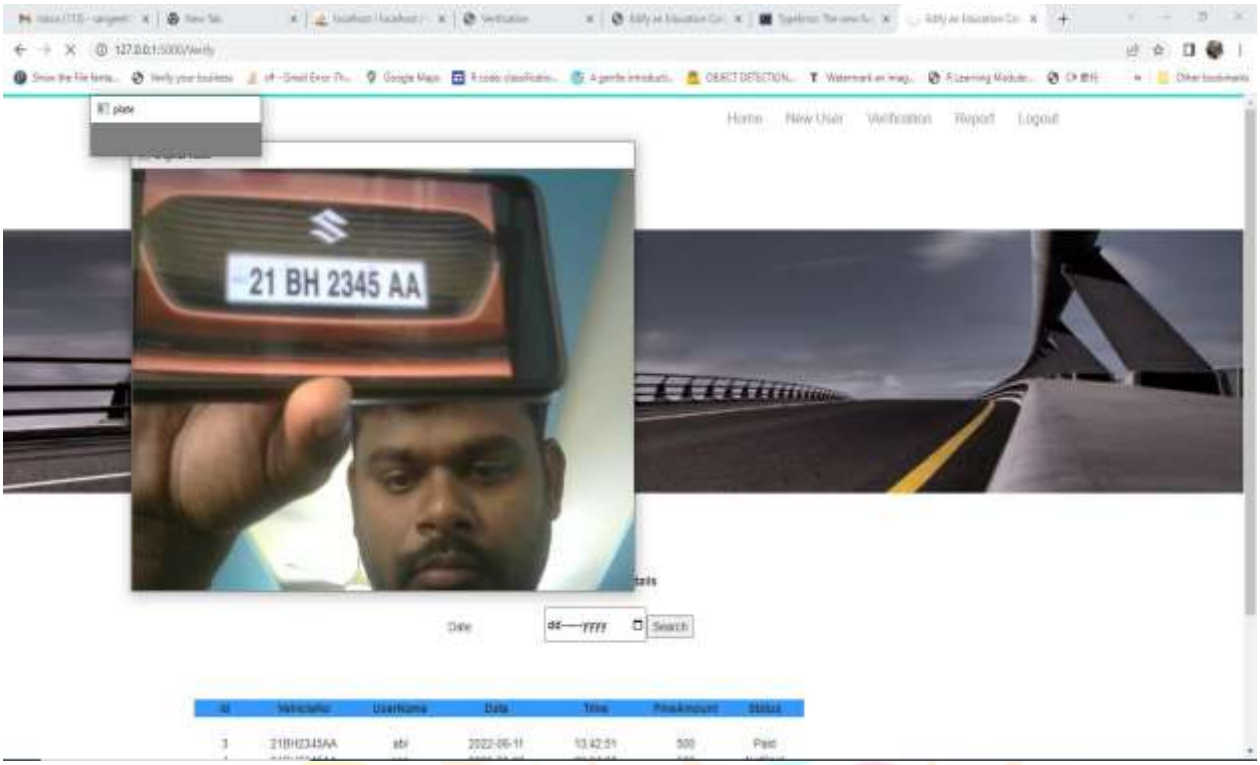


**Fig.7. Number Plate Recognition**

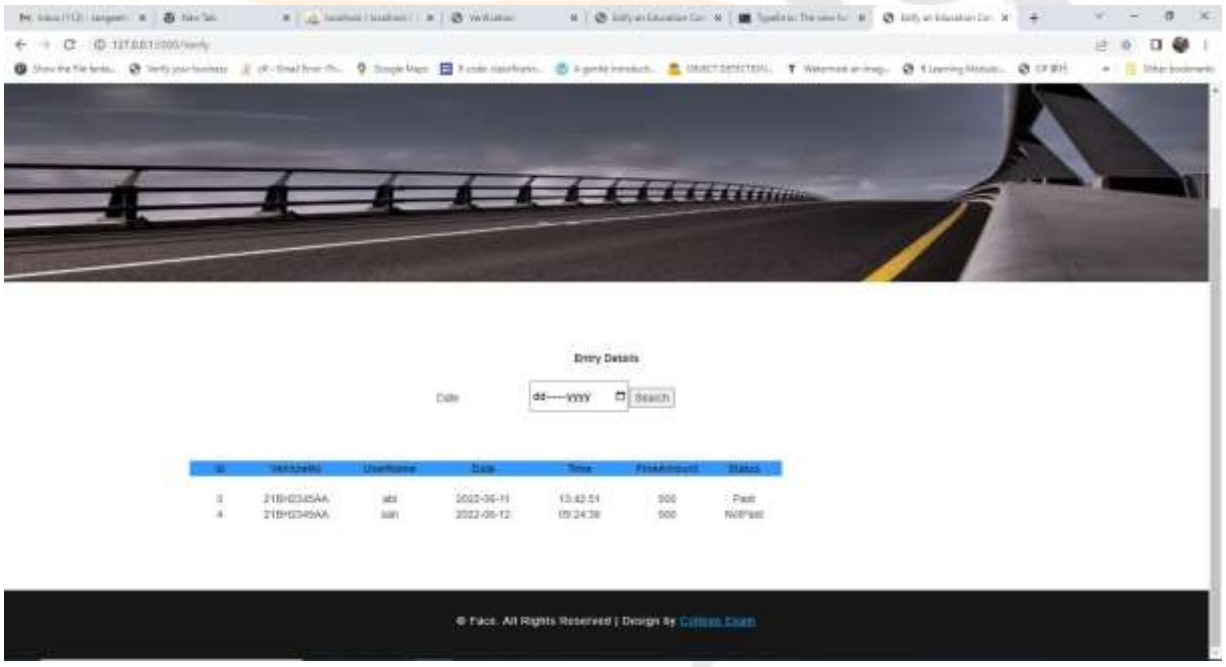


**Fig.8. Fine Details**

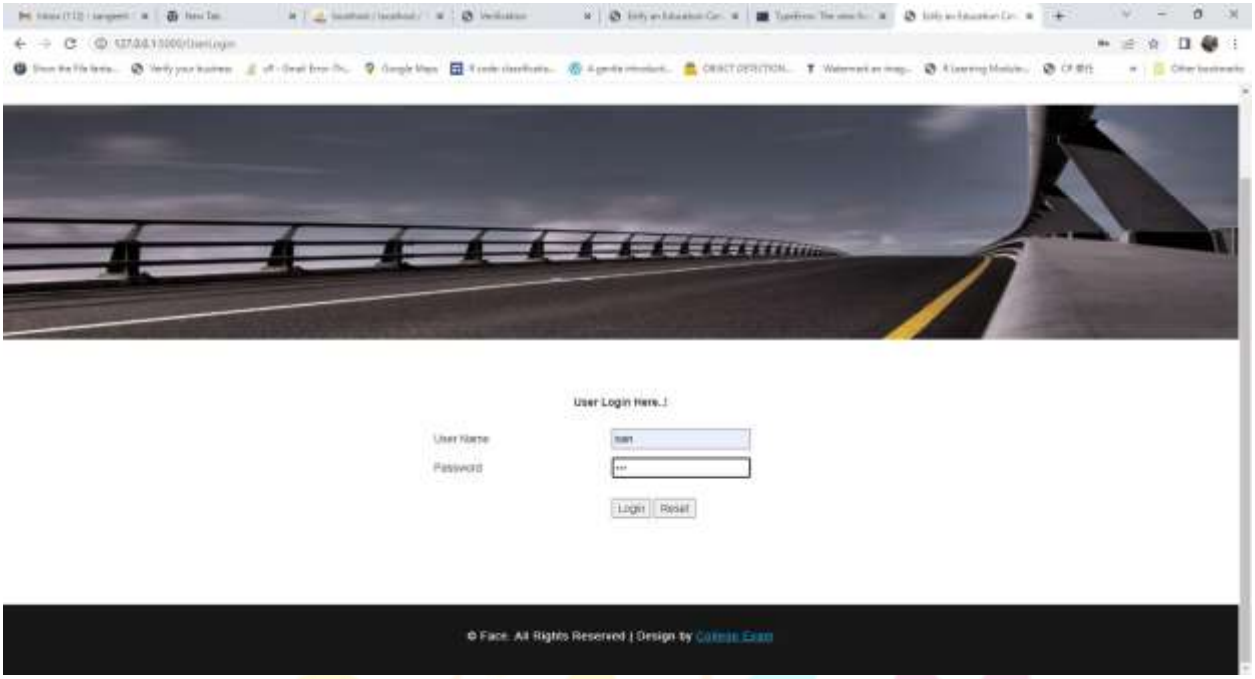


**Fig.9. User Login**

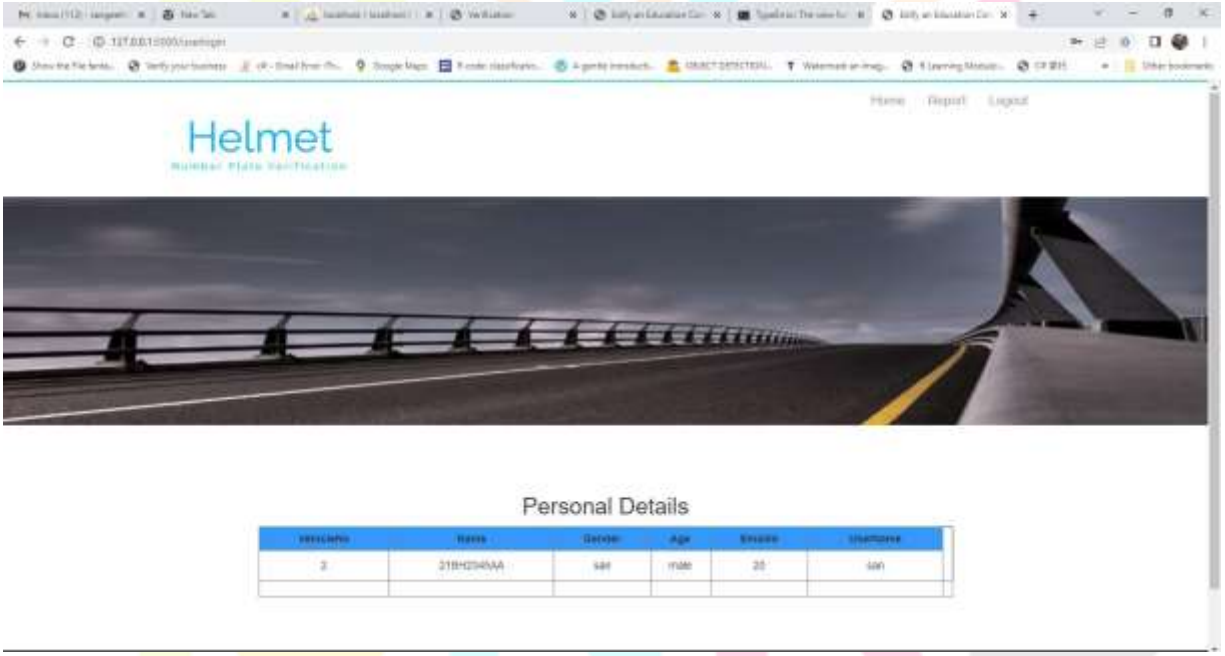


**Fig.10. Personal Details**

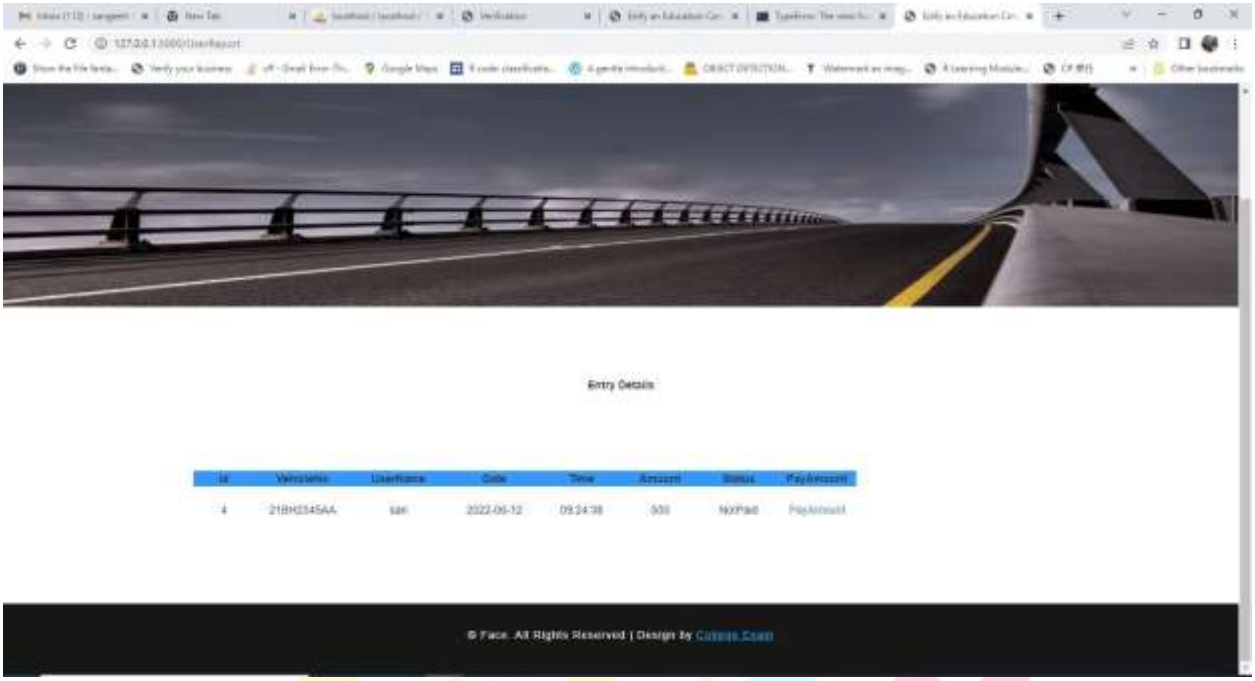


**Fig.11. Fine Entry Details**

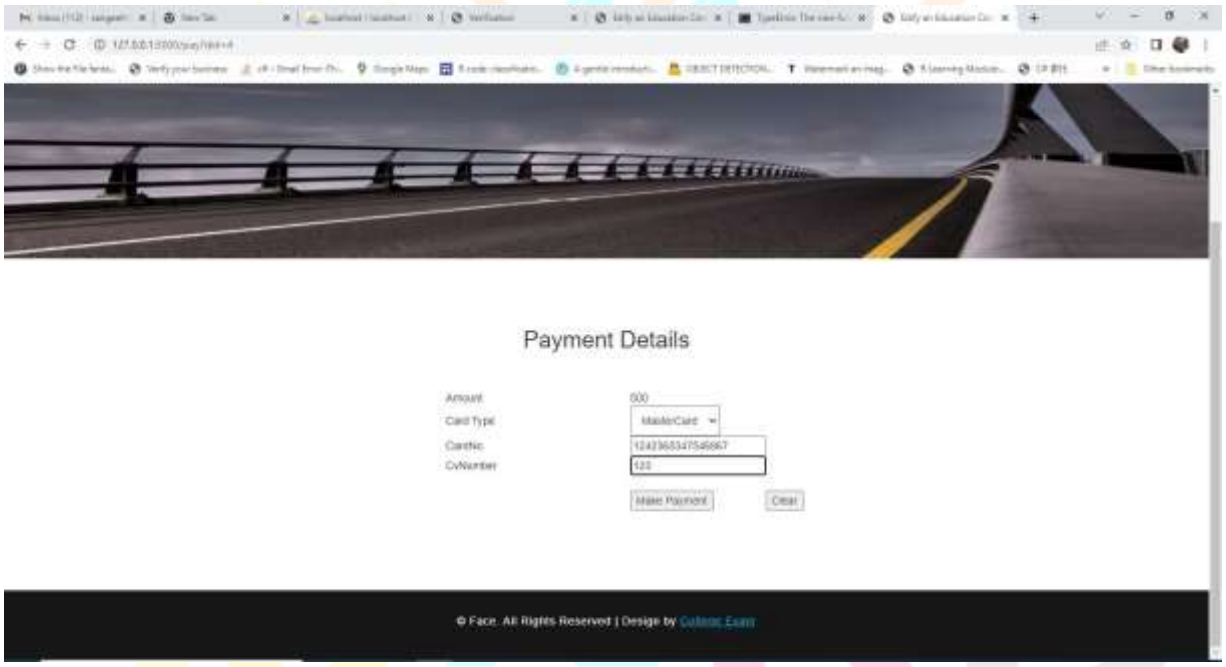


**Fig.12. Payment Details**

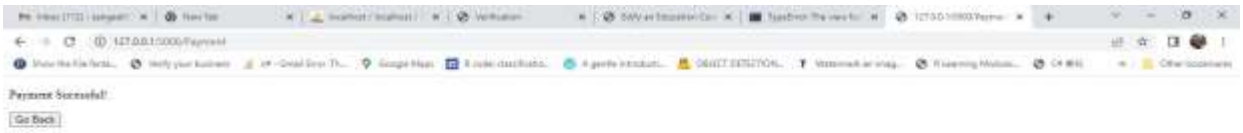


**Fig.13. Payment Status**

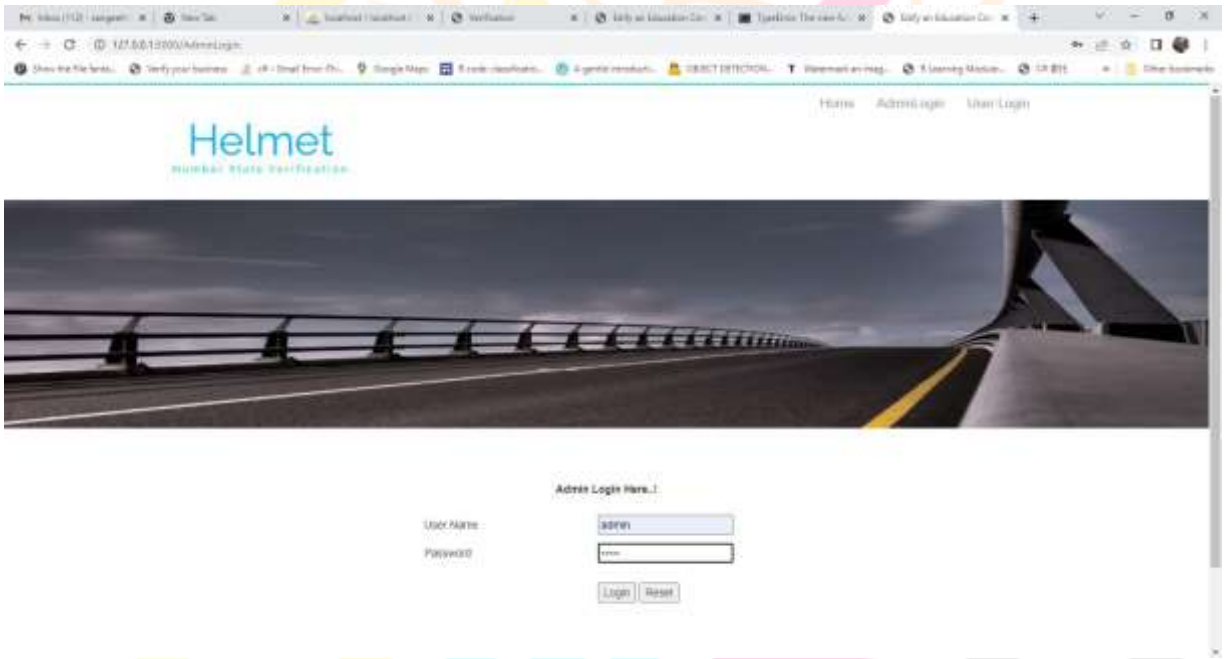


**Fig.14. Admin Page**

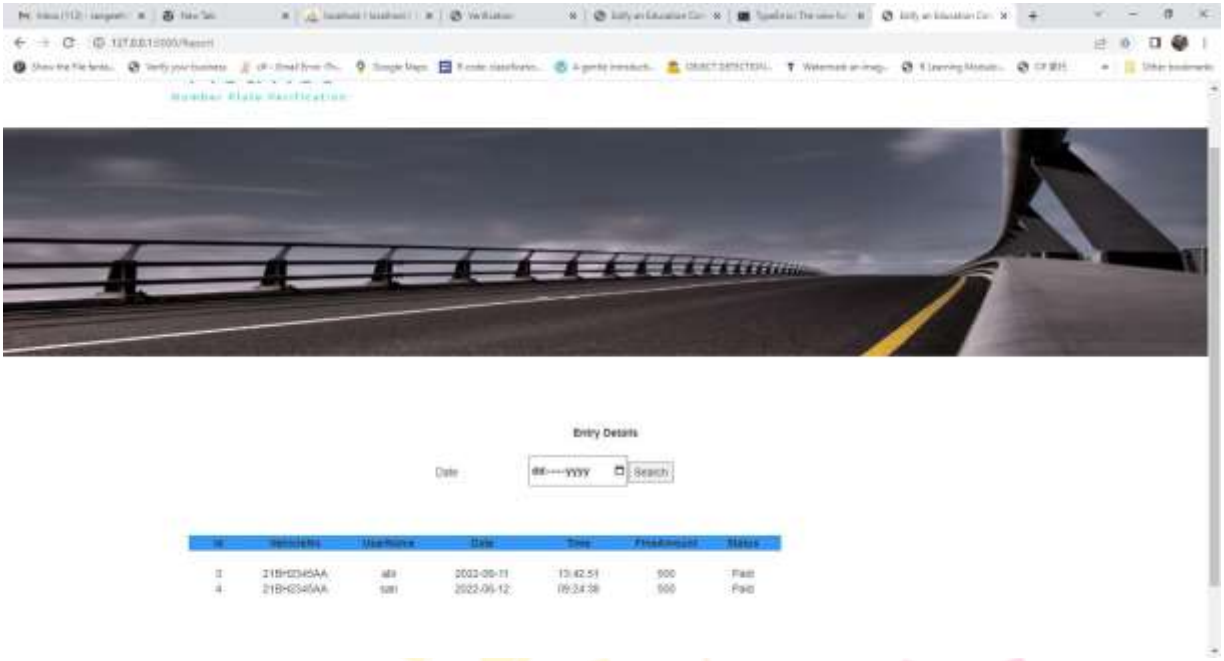


**Fig.15. Update Payment Status**

## REFERENCES

[1] Aboah, Armstrong, et al. "Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[2] An, Qing, et al. "Research on safety helmet detection algorithm based on improved YOLOv5s." Sensors 23.13 (2023): 5824.

[3] B. Lorch, S. Agarwal, and H. Farid, “Forensic Reconstruction of Severely Degraded License Plates,” Electronic Imaging, vol. 2019, no. 5, 2019.

[4] Chen, Junhua, et al. "Lightweight helmet detection algorithm using an improved YOLOv4." Sensors 23.3 (2023): 1256.

[5] Deng, Lixia, et al. "A lightweight YOLOv3 algorithm used for safety helmet detection." Scientific reports 12.1 2022.

[6] F. Schirrmacher, B. Lorch, B. Stimpel, T. Kohler, and C. Riess, “Sr ¨ 2 : Super-resolution with structure-aware reconstruction,” in 2020 IEEE International Conference on Image Processing (ICIP), 2020, p. 533537.

[7] G. Rossi, M. Fontani, and S. Milani, "Neural network for denoising and reading degraded license plates," in Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part VI. Springer International Publishing, 2021, pp. 484–499.

[8] Hayat, Ahatsham, and Fernando Morgado-Dias. "Deep learning-based automatic safety helmet detection system for construction safety." Applied Sciences 12.16 (2022): 8268.

[9] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated license plate recognition: a survey on methods and techniques," IEEE Access, vol. 9, pp. 11 203–11 225, 2020.

[10] M. Zhang, W. Liu, and H. Ma, "Joint license plate super-resolution and recognition in one multi-task gan framework," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.

[11] "Multitask learning," Machine learning, vol. 28, no. 1, pp. 41–75, 1997.

[12] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in ICML, 1993.

[13] S. Ruder, "An overview of multi-task learning in deep neural networks," arXiv preprint arXiv:1706.05098, 2017.

[14] Song, Hongru. "Multi-scale safety helmet detection based on RSSE- YOLOv3." Sensors 22.16 (2022): 6061.

[15] Susa, Julie Ann B., et al. "An efficient safety and authorized helmet detection using deep learning approach." 2022 International Conference on Smart Information Systems and Technologies (SIST). IEEE, 2022.

[16]Tran, Duong Nguyen-Ngoc, et al. "Robust automatic motorcycle helmet violation detection for an intelligent transportation system." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[17]Tsai, Chun-Ming, et al. "Video analytics for detecting motorcyclist helmet rule violations." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[18] Villar-Corrales, F. Schirrmacher, and C. Riess, “Deep learning architectural designs for super-resolution of noisy images,” in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 1635–1639.

[19] Waris, Tasbeeha, et al. "CNN-Based Automatic Helmet Violation Detection of Motorcyclists for an Intelligent Transportation System." Mathematical Problems in Engineering 2022 (2022).

[20] Y. Lee, J. Lee, H. Ahn, and M. Jeon, “Snider: Single noisy image denoising and rectification for improving license plate recognition,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Oct 2019.