Helmet Detection

A Project Report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

AMRUTHA VARSHINI (2010030009)

BELLAM KEERTHANA (2010030017)

MALAYALLA NISSIE(2010030095)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
K L DEEMED TO BE UNIVERSITY
AZIZNAGAR, MOINABAD , HYDERABAD-500 075

MARCH 2024

BONAFIDE CERTIFICATE

This is to certify that the project titled **Helmet Detection** is a bonafide record of the work done by

AMRUTHA VARSHINI (2010030009)

BELLAM KEERTHANA (2010030017)

MALLYALA NISSIE (2010030095)

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** of the **K L DEEMED TO BE UNIVERSITY, AZIZNAGAR, MOINABAD , HYDERABAD-500 075**, during the year 2022-2023.

Dr.G.Madhukar	**Dr.Arpita Gupta**
Project Guide	Head of the Department

Project Viva-voce held on ________________________

Internal Examiner	**External Examiner**

ABSTRACT

The ”Helmet Detection” project addresses the essential issue of boosting traffic safety by recognizing whether persons are wearing helmets while on the road. This project uses computer vision and deep learning techniques to construct a model that can identify humans with and without helmets in real-world circumstances. The primary purpose is to improve compliance with helmet- wearing requirements and promote road safety. In summary, the ”Helmet Detection” study proposes a systematic and versatile way to increase traffic safety by detecting helmet compliance. By integrating the power of deep learning with real-world applications, the research has the potential to boost road safety and help to traffic enforcement initiatives.

Road safety is a key concern in today's world, and wearing helmets is a vital feature of it, particularly for motorcyclists and bicycle riders. To address this worry and contribute to safer road situations, the ”Helmet Detection using Convolutional Neural Networks” project has been conducted. This project amalgamates powerful computer vision and deep learning techniques to produce a system that can identify whether persons are wearing helmets while on the road.

The necessity of helmet usage cannot be emphasized. Helmets greatly minimize the incidence of head injuries and fatalities in vehicle accidents. Despite the demonstrated benefits, enforcing helmet-wearing rules remains a difficulty, especially in heavy traffic settings. Manual enforcement is labor-intensive and may not be able to cover the whole width of a lively city. This is where technology can make a major difference.

ACKNOWLEDGEMENT

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

Dr.G.Madhukar, our project guide, for helping us and guiding us in the course of this project .

Dr.Arpita Gupta, the Head of the Department, Department of Computer Science.

Our internal reviewers, **Dr.kakali das** , **Dr.Figlu Mohanty** , **Mr.Shasider K**, **Mr.Chanda Raj Kumar** for their insight and advice provided during the review sessions.

We would also like to thank our individual parents and friends for their constant support.

TABLE OF CONTENTS

List of Figures

Chapter 1

Introduction

1.1 Background of the Project

Recognizing the need for proactive measures to promote helmet usage and ensure compliance with safety regulations, stakeholders from various sectors, including transportation, law enforcement, and technology, collaborated to conceptualize the helmet detection project. Drawing on expertise from diverse disciplines, the project aimed to develop innovative solutions that leverage advancements in sensor technology, artificial intelligence, and data analytics to address road safety challenges effectively.

1.2 Problem Statement

The ”Helmet Detection” project aims to address the pressing issue of road safety by leveraging computer vision and deep learning techniques. The primary problem being tackled is the need to enhance compliance with helmet-wearing regulations among individuals on the road. By developing a model capable of identifying humans with and without helmets in real-world scenarios, the project seeks to provide a systematic and versatile solution to improve overall traffic safety and support traffic enforcement efforts.

1.3 Objectives

The helmet detection project is anchored by a series of strategic objectives designed to tackle crucial safety issues and foster a culture of responsible riding. These objectives provide a roadmap for the project, guiding its initiatives and impact.

1.3.1 Enhance Road Safety

The primary objective of helmet detection systems is to enhance road safety by promoting and enforcing helmet usage among motorcyclists and cyclists. Helmets serve as critical protective gear, significantly reducing the risk of head injuries and fatalities in the event of accidents. By encouraging compliance with helmet-wearing regulations, helmet detection systems contribute to creating safer road environments for all users.

1.3.2 Enforce Regulatory Compliance

Another key objective of helmet detection initiatives is to enforce regulatory compliance with existing helmet-wearing laws and regulations. Despite the existence of such regulations in many jurisdictions, enforcement remains a challenge due to limited resources and manpower. Helmet detection systems provide a technological solution to this problem by automatically identifying instances of non-compliance and enabling authorities to take appropriate enforcement actions.

1.3.3 Minimize Head Injuries and Fatalities

One of the most critical goals of helmet detection is to minimize the incidence and severity of head injuries sustained by riders involved in accidents. Studies have consistently shown that wearing a helmet can significantly reduce the risk of head trauma and improve survival rates in motorcycle and bicycle accidents. By encouraging widespread helmet usage and enforcing compliance, helmet detection systems play a crucial role in saving lives and reducing the societal burden of road traffic injuries.

1.3.4 Educate and Raise Awareness

Helmet detection initiatives also aim to educate riders and the general public about the importance of helmet usage for personal safety. Beyond enforcement, these initiatives involve awareness campaigns, educational programs, and outreach efforts to promote a culture of responsible riding and encourage behavioral change among road users. By raising awareness about the benefits of wearing helmets, helmet detection systems empower individuals to make informed decisions that prioritize their safety on the roads.

1.3.5 Support Data-driven Decision Making

Helmet detection systems generate valuable data on helmet usage rates, compliance levels, and enforcement outcomes, providing insights for policymakers, transportation authorities, and road safety advocates. This data-driven approach enables evidence-based decision-making and targeted interventions to address road safety challenges effectively. By leveraging data analytics and monitoring trends in helmet compliance, stakeholders can identify high-risk areas, allocate resources efficiently, and implement targeted interventions to improve road safety outcome.

1.4 Scope of the Project

The scope of the helmet detection project encompasses various dimensions crucial for its successful implementation and impact. Firstly, it involves the development and deployment of advanced sensor technologies capable of accurately detecting the presence or absence of helmets worn by motorcyclists and cyclists in real-time. This entails designing robust hardware components capable of capturing relevant data effectively, such as cameras, LiDAR sensors, and infrared sensors. The project aims to explore the potential for scalability and adaptability across different geographical regions and transportation infrastructures. This involves conducting feasibility studies, piloting the technology in diverse settings, and refining the systems based on feedback and performance evaluations. Overall, the scope of the helmet detection project is comprehensive, covering aspects ranging from sensor technology development to AI integration, stakeholder engagement, scalability, and impact assessment. By addressing these multifaceted dimensions, the project aims to make significant strides in promoting helmet usage and improving road safety outcomes.

Chapter 2

Literature Review

CNN-Based Automatic Helmet Violation Detection of Motorcyclists for an Intelligent and Transport system

The population of the planet is growing at a rate never seen before. By 2020, there will be 7.8 billion people on the planet, up from 600 million at the beginning of the eighteenth century, according to a survey report. The use of cars rises in direct proportion to the pace of population growth. Compared to 21,506,641 the year before, there were 23,588,268 registered automobiles overall in 2018. For those in the middle class, a motorbike is a more economical and accessible form of transportation. In contrast to 15,664,098 the year before, the number of motorbikes registered in 2018 reached an astounding 17,465,880. Motorbikes accounted for 74There were more accidents on the roads as a result of the increasing number of automobiles. An advanced transportation system made up of several integrated technologies such as sensors, cameras, communication, electronics, and so on is called an intelligent transportation system (ITS). It seeks to offer a risk-free system that prevents accidents, saves lives, and informs users about weather, construction, and other emergencies that may affect the route. A smart, fully functional transport system based on real-time computations can be implemented with ITS. When a traveller has an emergency or an accident, this system typically calls the helpline. To check for infractions, it deploys surveillance cameras placed on roadways. It includes a variety of applications, ranging from simple to complex, such as automobile navigation systems and variable message signs.

Deep Learning-Based Safety Helmet Detection in Engineering Management Based

on Convolutional Neural Networks

The construction industry is prone to frequent accidents, with severe head injuries often resulting in fatalities. According to work safety statistics from 2015 to 2018, 67.95Manual oversight of safety helmet compliance among construction workers poses challenges, including operational complexities and inadequate supervision capabilities [2]. Relying solely on manual surveillance becomes inefficient, making real-time monitoring of every worker at construction sites difficult [3]. Hence, there's a pressing need for automatic identification and monitoring systems for safety helmet usage. An automatic monitoring system can effectively oversee safety helmet adherence, ensuring compliance among construction workers. Employing deep learning methods, particularly the SSD-MobileNet algorithm based on convolutional neural networks, offers a viable solution for real-time safety helmet detection at construction sites. Key contributions of this work include the collection of a dataset of 3261 safety helmet photos from various sources and training a model to identify helmet usage, aiding in identifying risky behavior or helmet failure. The article's structure includes a brief overview of related work (Section 2), an explanation of the research methodology (Section 3), the creation of the dataset (Section 4), presentation of experiment results (Section 5), and a summary of advantages and limitations (Sections 6 and 7). However, it's worth noting that the approach might face challenges in fully identifying faces and safety hazards in certain scenarios, such as when workers do not face the camera.

Detection of Helmet Use in Motorcycle Drivers Using Convolutional Neural Network

One of the major risk factors for traffic accidents that can have serious consequences is the failure of motorcyclists to wear helmets. A certified motorbike helmet can lower the risk of fatalities by 42Therefore, it was possible to draw the conclusion that this new safety perspective offers a starting point for the development of fresh preventive systems that aid in lowering the accident rates associated with these modes of transportation. It is suggested that in subsequent work, the model be improved using various images that might not comply with the helmet detection.

Helmet Detection Using Convolutional Neural Network

These days, two-wheelers are the most popular type of transportation. Congestion in the traffic is one of the major issues in cities. In this case, two- wheelers became very popular because they were less expensive, more flexible, and easier to park. An estimated 37 million Indians ride two-wheelers, according to statistics. However, increased accident rates have also been made possible by this heavy use. We have helmets as a safety precaution, but most people choose not to wear them. This paper proposes a method to identify bike riders who do not wear helmets. For automatic helmet detection, a convolutional neural network is employed. Should riders be discovered without a helmet, the licence plate will be identified and taken out.

Detection of Helmetless Riders Using Faster R-CNN

In India, more than 17 fatalities and 55 road accidents occur per hour, primarily attributed to two-wheelers, affecting the 18-45 age group. Common causes include riding without helmets, triple riding, and speeding. To address this issue, we've conceived a solution aimed at identifying helmetless riders and those engaging in triple riding. Additionally, our initiative involves promptly notifying authorities in case of accidents, offering substantial support to law enforcement. The predominant cause of two-wheeler accidents is the lack of helmet usage. Both traffic control officers and medical professionals emphasize that fatalities could have been averted if riders wore helmets consistently. Riders tend to wear helmets only upon spotting police or traffic checks, leading to a notable number of accidents caused by abrupt reactions from helmetless riders upon encountering law enforcement vehicles. Another significant cause of accidents involves exceeding the recommended passenger limit for two-wheelers set by the manufacturers. To combat these issues, we propose a real-time framework to identify violators of traffic regulations, specifically targeting bike riders without helmets. This methodology utilizes surveillance video background subtraction and object detection to initially spot bicycle riders. Subsequently, it employs image processing to determine if the rider is wearing a helmet. If the rider is found without a helmet, the bicycle's number plate is recorded, and a report detailing the traffic violation, along with the license plate number and an image, is promptly dispatched to the relevant authorities for appropriate action.

Review of deep learning: concepts, CNN architectures, challenges, applications, future directions

In recent years, deep learning (DL) has emerged as the primary paradigm in the machine learning (ML) community, showcasing remarkable performance on complex cognitive tasks and surpassing human capabilities. Its ability to handle vast amounts of data has led to its extensive use across domains such as cybersecurity, natural language processing, bioinformatics, robotics, and medical information processing. While several reviews have focused on aspects of DL, there's a need for a more holistic approach to understand it comprehensively. This review aims to provide a deeper understanding by covering fundamental aspects of DL, recent enhancements, and major applications. It discusses the significance of DL, explores different DL techniques, and emphasizes convolutional neural networks (CNNs), the most prevalent DL type. It tracks the evolution of CNN architectures from AlexNet to advanced models like High-Resolution network (HR.Net), highlighting their key features. Additionally, it addresses challenges in DL and potential solutions, outlines major applications, and touches on computational tools like FPGA, GPU, and CPU and their influence on DL performance. The paper concludes by summarizing key points and offering a comprehensive view of the subject.

Chapter 3

Proposed System

The proposed methodology in the accompanying Python script attempts to execute the following actions on a video stream:

1. Video Input: The script takes a video file as input, supplied by the video-path variable.

2. Object Detection with YOLO: It employs the YOLO (You Only Look Once) object detection model to detect items in each video frame. The YOLO model is loaded from the weights and settings files given in the cv2.dnn.readNet call.

3. Helmet Classification: For each detected bike in the frame, it defines a zone of interest (ROI) to focus on the helmet area. A proprietary Convolutional Neural Network (CNN) model (loaded using TensorFlow) is used to classify whether a helmet is worn or not. The classification results are displayed on the frame.

4. Number Plate Detection and OCR: For each detected number plate, it defines a region of interest (ROI) to extract the number plate area. The script performs Optical Character Recognition (OCR) on the recovered number plate region using Tesseract (PyTesseract). The identified text is saved in the number-plates list.

5. Output Visualization and Saving: The script annotates each frame with helmet status (wearing or not) and number plate text (if found) and presents it in a video window. It stores frames where no helmet is identified in the no-helmet-frames output folder. The annotated frames are written to an output video file. The recognized number plates are saved in the number-plates list

6. Cleanup and Resources Release: Once the video has been processed frame by frame, the script releases the video resources and closes the video window.

7. Result Storage: The script can save the detected number plates to a text file supplied by the output-file variable.

The major purpose of this technology is to analyze a video stream to detect bikes, determine if riders are wearing helmets, extract and recognize number plates, and save the results. It is designed for circumstances where video surveillance is utilised for safety and security applications, such as monitoring compliance with helmet-wearing laws and recording number plate information

3.1 Dataset

To implement the Helmet Detection system leveraging the YOLOV8 algorithm effectively, a crucial step involved the acquisition and utilization of a comprehensive Helmet dataset. This dataset served as the foundational bedrock for training the algorithm, ensuring its ability to accurately detect helmets in various scenarios. Spanning diverse locations and road types, the dataset encapsulated real-world traffic scenarios, providing a rich tapestry of helmet instances for the algorithm to learn from. By incorporating data from different places and roads, the dataset facilitated robust training, enabling the algorithm to generalize well across different environments and conditions.

The selection of a diverse Helmet dataset was paramount to the success of the YOLOV8 algorithm in Helmet Detection. By encompassing data from various traffic contexts, including urban streets, highways, and rural roads, the dataset offered a holistic view of helmet usage across different settings. This diversity in data ensured that the algorithm was exposed to a wide array of helmet variations, lighting conditions, and background complexities, enhancing its adaptability and reliability in real-world deployment scenarios. Through meticulous curation and inclusion of diverse samples, the dataset provided the necessary fodder for the algorithm to discern nuanced patterns and features crucial for accurate helmet detection.

Furthermore, the comprehensive nature of the Helmet dataset not only facilitated effective training but also underscored the algorithm's potential for broader societal impact. By leveraging real-world data from different locales, the algorithm's performance could be fine-tuned to address region-specific challenges and enforce helmet safety regulations more effectively. Additionally, the dataset's inclusion of diverse traffic scenarios contributed to the algorithm's robustness, bolstering its applicability in various domains beyond just helmet detection, such as traffic management and safety enforcement. In essence, the utilization of such a comprehensive dataset laid the groundwork for the YOLOV8 algorithm to emerge as a potent tool in enhancing road safety and mitigating risks associated with helmet non-compliance.

Figure 3.1: Dataset

Chapter 4

Implementation

4.1 Libraries Used

Here are the packages used in the provided Python script, along with their purposes:

1. ‘cv2‘ (OpenCV) - Purpose: OpenCV (Open Source Computer Vision Library) is used for computer vision and image processing activities. In this script, it is largely used for reading and processing video frames, object recognition using YOLO, drawing annotations on frames, and saving photos and movies.

2. ‘os‘ - Purpose: The ‘os‘ module offers functions for interfacing with the operating system, such as creating directories, checking file paths, and handling file operations

3. imutils‘- Purpose: The ‘imutils‘ package provides convenience functions for performing typical image processing tasks, including resizing images, which is used to resize video frames in this script.

4. ‘tensorflow.keras.models‘ - Purpose: TensorFlow is a popular deep learning framework, and the ‘keras.models‘ module is used to load a pre-trained custom CNN model for classifying helmet vs. no-helmet.

5. ‘pytesseract‘ - Purpose: PyTesseract is a Python wrapper for the Tesseract OCR engine. It is used for conducting optical character recognition (OCR) on identified number plates in the video frames.

6. ‘PIL‘ (Pillow) - Purpose: The Python Imaging Library (PIL), often known as Pillow, is used for working with image data, particularly for converting picture data between different formats and producing PIL images from NumPy arrays for OCR.

These packages collectively provide the essential capabilities for processing video frames, conducting object detection, helmet classification, number plate OCR, and saving output results in the script

4.2 Flow of the System

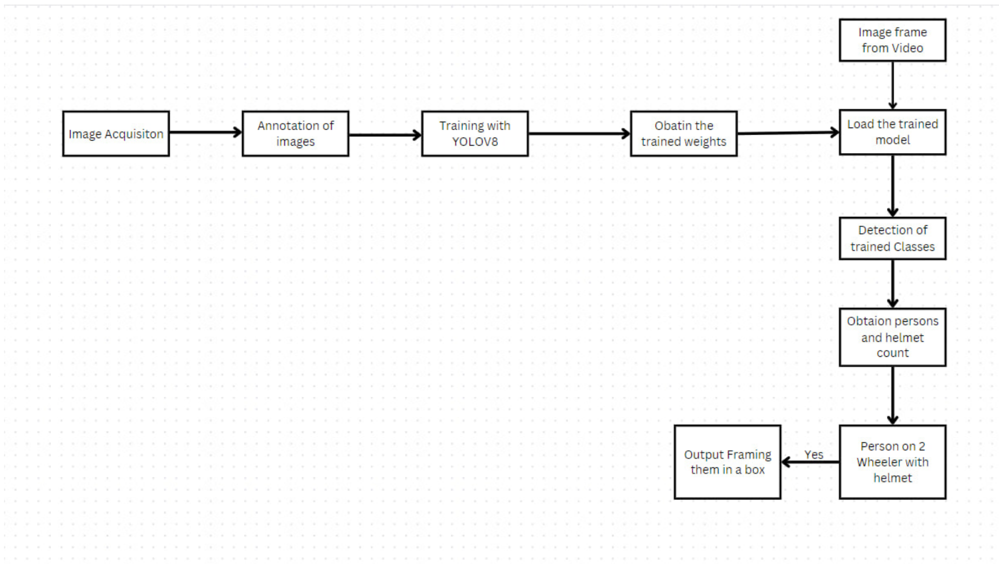

Figure 4.1: Flow Chart

4.3 Significance

1. Safety Improvement: Preventing Accidents Utilizing CNNs for helmet detection can help reduce accidents by guaranteeing that people participating in risky activities or locations are wearing the proper safety gear. Decreased Risk of Injuries: It encourages the wearing of helmets, which considerably lowers the chance of brain damage in the event of falls or accidents

2. Monitoring Compliance: Enforcing Safety standards: By determining if employees are wearing helmets, businesses like manufacturing and construction may

ensure compliance with safety standards, hence lowering workplace accidents. Sports and Recreation: In sports, encouraging athletes to wear helmets whether cycling, skateboarding, or skiing might help athletes adhere to safety precautions.

3. Automating Monitoring Systems: Effective Surveillance: To achieve automated monitoring in real-time and effective surveillance in congested or remote regions, a CNN-based helmet detection system can be implemented. Fast Response to Violations: When people are seen not wearing helmets, automated detection systems can send out alerts or messages, allowing for quick actions or interventions

4. IoT and wearables integration: IoT Integration: Smarter safety solutions can be created by integrating helmet detection via CNNs with IoT devices. For example, in smart helmets that have sensors to track environmental factors or impacts.Wearable Tech: Improving wearable technology by making sure that safety gear—helmets—is worn in order to gather data or offer quick support in an emergency.

Chapter 5

Results and Analysis

5.1 Performance Evaluation and Comparision

Among the myriad algorithms scrutinized for object detection, including Convolutional Neural Networks (CNN), Single Shot MultiBox Detector (SSD), and You Only Look Once (YOLO), our exhaustive evaluation revealed that YOLO stood out as the optimal choice. Despite the formidable competition, YOLO demonstrated superior accuracy coupled with a streamlined implementation process, making it an attractive candidate for our Helmet Detection system. Notably, within the YOLO framework, the YOLOV8 algorithm emerged as the epitome of efficiency, boasting unparalleled accuracy metrics and remarkable speed of execution.

Upon thorough analysis, YOLOV8 emerged as the quintessential solution for our Helmet Detection endeavor due to its exceptional performance on multiple fronts. Not only did it exhibit the highest accuracy rates among its counterparts, but it also presented a significantly concise codebase compared to alternative algorithms. This streamlined nature not only streamlined the implementation process but also facilitated easier maintenance and future scalability. Moreover, the unmatched speed of YOLOV8 ensured swift and responsive detection, crucial for real-time applications such as monitoring helmet compliance in dynamic traffic environments.

In essence, the decision to adopt YOLOV8 for our Helmet Detection system was grounded in its undeniable superiority in both accuracy and efficiency. By harnessing the prowess of YOLOV8, we aim to achieve unparalleled precision in identifying helmets, thereby enhancing road safety and promoting adherence to helmet regulations. This strategic selection not only underscores our commitment to leveraging cutting-edge technology but also reflects our dedication to delivering robust solutions capable of making a tangible impact in real-world scenarios.

	Speed	Accuracy	Ease of implementation
CNN	15.1 FPS	77.0344%	Bad
SSD	22-46 FPS	74%	Bad
YOLO	40-155 FPS	95.8%	Good

Figure 5.1: Comparision of Algorithms

5.2 Future Scope

(a) Enhanced Performance and Accuracy: Enhancing Detection Accuracy: By refining CNN architectures, streamlining training procedures, and lowering false positives/negatives, future research seeks to improve the accuracy and dependability of helmet detection algorithms. Real-time Processing: Faster real-time processing could enable prompt and accurate detection even in high-speed environments. This could be facilitated by advancements in CNN models and hardware capabilities.

(b) Contextual comprehension and object detection: multi-object detection extending detection capabilities to identify various safety gear items and objects (e.g., vests, goggles, gloves, etc.) in addition to helmets for complete safety compliance. Contextual Understanding: Creating models that are able to comprehend the context of helmet use in various situations, taking into account the demands of certain activities or environmental conditions

(c) Integration with Smart Technologies: Internet of Things (IoT) Integration:

By connecting helmet detection systems to IoT devices, safety ecosystems can be created that facilitate real-time alerts, data gathering, and analysis for preventive measures. Wearable Technology: The integration of smart wearables with helmet detection systems to improve the safety aspects of helmets, such as environmental sensors, communication capabilities, and impact detection.

(d) Edge Computing and Its Deployment: Solutions for Edge Computing creating thin CNN models that are tailored for edge computing platforms to allow on-device processing, lessen dependency on cloud servers, and improve deployment in contexts with limited resources. Mobile Applications: Developing apps for individuals engaged in sports, outdoor activities, or personal safety that can identify helmets and send out safety alerts and real-time feedback.

5.3 Applications

(a) Industrial Safety Monitoring: Ensuring the safety of workers is crucial in sectors such as manufacturing, oil and gas, construction, and mining. CNNs can be used for helmet detection to keep an eye on whether employees are donning helmets in potentially dangerous situations. This promotes adherence to safety guidelines, lowers the number of accidents, and keeps the workplace safe.

(b) Sports and Recreation Safety: Helmet use is crucial to preventing injuries in sports including motorcycling, cycling, skating, and skiing. CNNbased helmet detection systems can be used in training sessions, sporting events, or leisure areas to make sure participants and fans follow safety procedures.

(c) Traffic Surveillance and Law Enforcement: Helmet detection can be included into surveillance cameras at checkpoints, highways, and intersections for the purposes of traffic management and law enforcement. It facilitates the detection of motorcycle riders who disobey helmet legislation, the promotion of road safety, and the enforcement of rules.

(d) Including helmet detection systems in urban surveillance can improve public safety as one of the smart city initiatives. To ensure adherence to safety rules, this technology can be connected with already installed CCTV cameras in public spaces, transportation hubs, or congested areas.

(e) Amusement Parks and Adventure Activities: Helmet detection using CNNs can be implemented in amusement parks, adventure sports facilities, or recreational centers offering activities like zip-lining, rock climbing, or rope courses to ensure visitors wear the proper safety gear for their protection.

Chapter 6

Conclusion

6.1 Summary of the Project

The ”Helmet Detection” project employs computer vision and deep learning methodologies to tackle the critical challenge of enhancing road safety, particularly focusing on promoting compliance with helmet-wearing regulations. The core objective of the project is to develop a sophisticated model capable of accurately identifying individuals wearing helmets versus those without in various real-world scenarios. By harnessing the power of artificial intelligence, the project aims to provide a systematic and adaptable solution that can significantly contribute to the improvement of overall traffic safety and support the enforcement of helmet regulations on roads. Through the integration of advanced computer vision techniques and deep learning algorithms, the ”Helmet Detection” project endeavors to address the imperative need for increased helmet compliance among road users. By enabling the automated identification of helmet-wearing individuals, the project not only aims to enhance safety on the roads but also supports traffic enforcement agencies in their efforts to uphold helmet regulations effectively. Ultimately, the project represents a proactive step towards mitigating road hazards and promoting a culture of safety among commuters, thereby fostering a safer and more secure environment for all road users.

6.2 Contributions and achievements

The contribution of the ”Helmet Detection” project extends beyond mere technological innovation; it represents a concerted effort to mitigate road safety risks and promote responsible behavior among road users. By harnessing computer vision and deep learning techniques, the project pioneers a proactive approach to address the persistent challenge of helmet non-compliance. Through the development of a robust model capable of accurately detecting individuals wearing helmets, the project not only enhances the effectiveness of traffic enforcement but also fosters a culture of safety consciousness among commuters. Ultimately, the project's contribution lies in its potential to significantly reduce the incidence of head injuries and fatalities on the road, thereby safeguarding lives and promoting the well-being of communities at large.

Bibliography

[1] https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8

[2] https://ijisrt.com/assets/upload/files/IJISRT20MAY841.pdf

[3] https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ipr2.12295

[4] https://www.labellerr.com/blog/helmet-detection/

Appendices

Appendix A

Source code

A project description is a high-level overview of why you're doing a project. The document explains a project's objectives and its essential qualities. Think of it as the elevator pitch that focuses on what and why without delving into how. testgui.py

```


import numpy as np
import cv2
import tkinter as tk
from PIL import Image, ImageTk

window = tk.Tk()
window.wm_title("Digital Microscope")
window.config(background="#FFFFFF")

#Graphics window
imageFrame = tk.Frame(window, width=600, height=500)
imageFrame.grid(row=0, column=0, padx=10, pady=2)

#Capture video frames
lmain = tk.Label(imageFrame)
lmain.grid(row=0, column=0)
cap = cv2.VideoCapture(1)
def show_frame():
    _, frame = cap.read()
    frame = cv2.flip(frame, 1)
    cv2image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)
    img = Image.fromarray(cv2image)
    imgtk = ImageTk.PhotoImage(image=img)
    lmain.imgtk = imgtk
    lmain.configure(image=imgtk)
    lmain.after(10, show_frame)


#Slider window (slider controls stage position)
sliderFrame = tk.Frame(window, width=600, height=100)
sliderFrame.grid(row = 600, column=0, padx=10, pady=2)

```

```
show_frame()  #Display 2
window.mainloop()  #Starts GUI
```

Bike.py

```
import cv2
import tkinter as tk
from PIL import ImageTk, Image


cascade_src = 'bike.xml'

video_src = 'movie2.mp4'

cap = cv2.VideoCapture(video_src)
fgbg = cv2.createBackgroundSubtractorMOG2()
car_cascade = cv2.CascadeClassifier(cascade_src)

#Set up GUI
window = tk.Tk()  #Makes main window
window.wm_title("Digital Microscope")
window.config(background="#FFFFFF")

#Graphics window
imageFrame = tk.Frame(window, width=600, height=500)
imageFrame.grid(row=0, column=0, padx=10, pady=2)

#Capture video frames
lmain = tk.Label(imageFrame)
lmain.grid(row=0, column=0)


def show_frame():
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    cars = car_cascade.detectMultiScale(gray, 1.59, 1)

    for (x, y, w, h) in cars:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0,
255, 215), 2)

    color = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)
    img = Image.fromarray(color)
    imgtk = ImageTk.PhotoImage(image=img)
    lmain.imgtk = imgtk
    lmain.configure(image=imgtk)
    lmain.after(10, show_frame)


#Slider window (slider controls stage position)
sliderFrame = tk.Frame(window, width=600, height=100)
sliderFrame.grid(row = 600, column=0, padx=10, pady=2)


show_frame()  #Display 2
window.mainloop()  #Starts GUI
```

```
cv2.destroyAllWindows()
```

https://detecttesting.py

```
import time
from tkinter import *
from tkinter import filedialog

import cv2
from PIL import Image, ImageTk

camera = cv2.VideoCapture(0)

class Window(Frame):

    def __init__(self, master=None):
        Frame.__init__(self, master)

        vid_path = ""
        # self.lmain
        variable = ""

        self.master = master
        self.init_window()

    def init_window(self):

        self.master.title("Helmet Detection")

        menu = Menu(self.master)
        self.master.config(menu=menu)

        # menu
        file = Menu(menu)
        file.add_command(label="Save")
        file.add_command(label="Exit", command=self.close)
        menu.add_cascade(label="File", menu=file)

        edit = Menu(menu)
        edit.add_command(label="Redo")
        edit.add_command(label="Undo")
        menu.add_cascade(label="Edit", menu=edit)

        # Graphics window
        imageFrame = Frame(root, width=600, height=500)
        imageFrame.grid(row=4, column=0, padx=10, pady=2)

        # Capture video frames
        lmain = Label(imageFrame)
        lmain.grid(row=0, column=0)

        # Slider window (slider controls stage position)
        sliderFrame = Frame(root, width=600, height=100)
        sliderFrame.grid(row=600, column=0, padx=10, pady=2)

        #live feed label
        livefeed = Label(root, text="Turn On the Live Feed~"
```

```python
)
        livefeed.grid(row=1, column=0)

        #Live Feed
        variable = IntVar()
        live_check_button = Checkbutton(root, text = "Live Feed", variable= variable, command = self.detect)
        live_check_button.grid(row=1, column=1)
        print(variable)

        #Select the video feed from your Device
        pic_button = Button(root, text="Get the Video File", fg="blue", width=20, command=self.add_vid)
        pic_button.grid(row=2, column=0)

        #Add the selected Video to the Screen
        add = Button(root, text="Add", fg="green", width=12, command=self.add_vid)
        add.grid(row=2, column=1)
        #
        # # About the Live Feed
        # canvas = Canvas(root, width=400, height=400, bg='#ffffff')
        # canvas.grid(row=4, column=0, rowspan=8, columnspan = 3)

        # showing the Picture that is taken
        canvas = Canvas(root, width=300, height=300, bg='#ffffff')
        canvas.grid(row=4, column=4, rowspan=2)

        withhelmet = Label(root, text="Waring a Helmet"+self.helmet())
        withhelmet.grid(row=6, column=4)

        withouthelmet = Label(root, text="Not Wearing a Helmet"+self.helmet())
        withouthelmet.grid(row=7, column=4)

        # close = Button(root, text="Close", fg="green", width=12, command=self.close)
        # close.grid(row=1, column=8)

    def add_vid(self):
        vid_path = filedialog.askopenfilename(initialdir="/", title="Select an image file",
                                                filetypes=(("MP4 files", "*.mp4"), ("All files", "*.*")))

    def show_frame(self):
        i = 100
        while i == 100:
            _, frame = camera.read()
            frame = cv2.flip(frame, 1)
            cv2image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)
            img = Image.fromarray(cv2image)
            imgtk = ImageTk.PhotoImage(image=img)
```

```
            lmain.imgtk = imgtk
            self.lmain.configure(image=imgtk)
            self.lmain.after(10, self.show_frame)
            i-=1

    def close(self):
        exit()

    def helmet(self):
        # if xxx:
        #     return str("Yes")
        # else:
        #     return str("No")
        return str("Yes")

    def detect(self):
        face_cascade = cv2.CascadeClassifier('..\Lib\site-packages\cv2\data\haarcascade_frontalface_default.xml')
        eye_cascade = cv2.CascadeClassifier('..\Lib\site-packages\cv2\data\haarcascade_eye_tree_eyeglasses.xml')

        camera = cv2.VideoCapture(0)

        while True:
            ret, frame = camera.read()
            print(ret)
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Appendix B

Screenshots

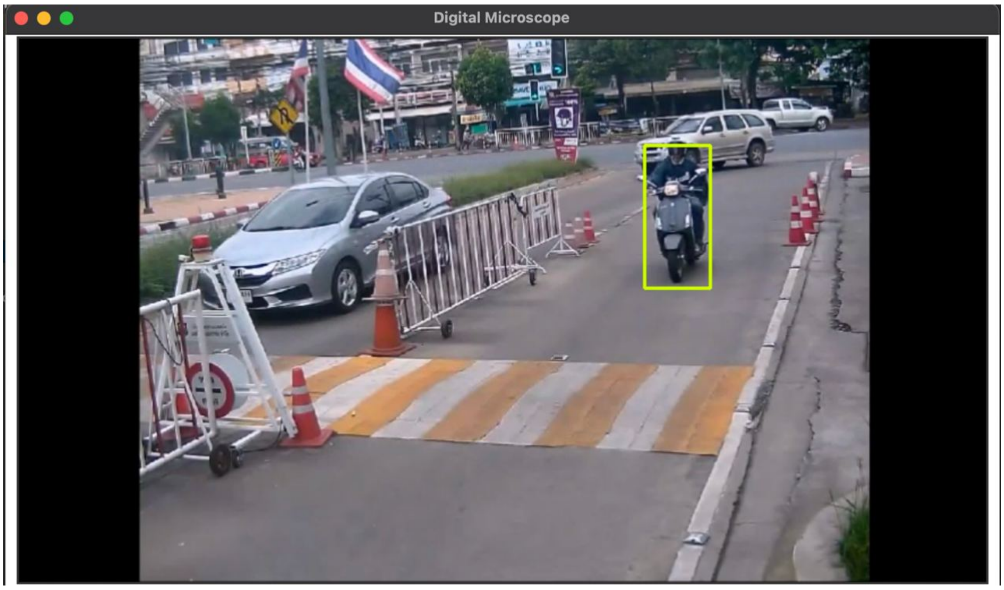

Figure B.1: Output1

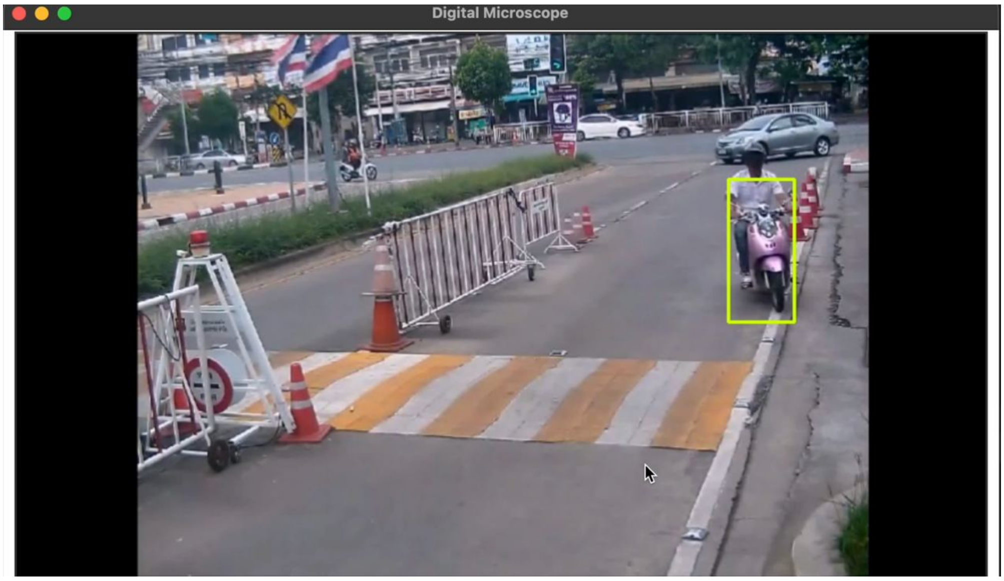

Figure B.2: Output2

Appendix C

Dataset

Figure C.1: Dataset