TRAFFIC ACCIDENT DETECTION SYSTEM USING DEEP LEARNING

FINAL YEAR PROJECT

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

BACHELOR OF SCIENCE

IN

COMPUTER TECHNOLOGY

Submitted by

SRIVATSAN MK

22BCT049

Under the Guidance of

Dr. G. Sumalatha, MCA., M.Phil., Ph.D.,

Assistant Professor

Department of Computer Technology and Data Science

SRI KRISHNA ARTS AND SCIENCE COLLEGE

(An Autonomous Institutions Affiliated to Bharathiar University)

Accredited By NAAC with 'A' Grade

Kuniyamuthur, Coimbatore - 641008

MARCH 2025

SRI KRISHNAARTS AND SCIENCE COLLEGE

Accredited By NAAC with ’A’ Grade

Kuniyamuthur, Coimbatore - 641008

DECLARATION

I hereby declare that the Project Report entitled **“TRAFFIC ACCIDENT DETECTION SYSTEM USING DEEP LEARNING”** submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Science in Computer Technology** is an original work submitted and it has not been previously formed the basis for the award of any other Degree, Diploma, Associate ship, Fellowship or similar titles to any other university or body during the period of my study.

Place: Coimbatore

Date:

SIGNATURE OF THE STUDENT

SRI KRISHNA ARTS AND SCIENCE COLLEGE

Accredited By NAAC with 'A' Grade

Kuniyamuthur, Coimbatore - 641008

CERTIFICATE

This is to certify that the Project Report entitled **"TRAFFIC ACCIDENT DETECTION SYSTEM USING DEEP LEARNING"** in partial fulfilment of requirements for the award of the degree of Bachelor of Science in Data Science is a record of bonafide work carried out by **SRIVATSAN MK (22BCT049)** and that no part of this has been submitted for the award of any other degree or diploma and the work has not been published in popular journal or magazine.

GUIDE **HoD** **DEAN**

This Project Report is submitted for the viva voce conducted on ____________________at Sri Krishna Arts and Science College.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I am ineffably indebted to **Dr. K. Sundararaman, M.Com., M.Phil., Ph.D., Chief Executive Officer**, Sri Krishna Institutions, Coimbatore.

I convey my profound gratitude to **Dr. R. Jagajeevan, MBA., M.Phil., Ph.D., Principal**, Sri Krishna Arts and Science College for giving permission for the fulfilment of the venture.

It is my prime to solemnly express my sense of gratitude to **Dr. K. S. Jeen Marseline, MCA., M.Phil., Ph.D., Dean, Computer Stream and Mathematics**, Sri Krishna Arts and Science College.

It is my prime to solemnly express my sense of gratitude to **Dr. B. Radha, MCA., Ph.D., Head & Associate Professor, Department of Computer Technology and Data Science**, Sri Krishna Arts and Science College.

I would like to extend my thanks and unbound sense for the timely help and assistance given by **Dr. G. Sumalatha, MCA., M.Phil., Ph.D., Assistant Professor, Department of Computer Technology and Data Science**, Sri Krishna Arts and Science College in completing the Final Project work. Her remarkable guidance at every stage of my work was coupled with suggestion and motivation.

I take this opportunity to thank my parents and friends for their constant support and encouragement throughout this training.

SRIVATSAN MK

22BCT049

TABLE OF CONTENT

ORGANIZATION PROFILE

ORGANIZATION NAME: INSOURCE TECHNOLOGIES

Insource was established on November 2008 by Mr. S.ArunKumar, Managing Director and Founder of the company and Mr.A.Farook as the Management Trustee. Insource group offers the entire enterprise solutions under one roof. Our group includes Insource Technology for IT and automobile Solutions, Day2Day Ads for Graphic & Print Solutions. We have been successfully leading in the business arena of software development services, Embedded ,web design and web development services with aggressive internet marketing campaigns for top search engine rankings; Graphic & Print design services and Hardware and software solutions all at one shop.

We prepared our self patently to ref left the vision and the goal of your organization. Since our customer satisfaction is our satisfaction, we do not measure success in numbers but by the level of satisfaction. Now a new dimension from Insource group, Insource InfoTech has developed by the Insource team in June 2010 for Hardware Procurements. Insource InfoTech is a topnotch shop for buying all Hardware and Software procurements. We provide Desktop Computers, Laptops, Workstations & Servers, Printers, Networking Equipments, LCD Projectors, surveillance cameras and many more hardware and software related procurements for reasonable price.

Mission: To provide customized and quality service to our customers, and improve achievement of goals and profitability.

Vision: To become a leading service-providing company aimed at mutual growth among the company, employees, and customers, without compromising quality.

ABSTRACT

Accurate and timely detection of road accidents is crucial for enhancing road safety and minimizing potential hazards. In this project, we present an advanced Accident Detection Model based on a state-of-the-art real-time object detection framework. Our model utilizes computer vision techniques to analyze road surveillance videos, identify potential accidents, and provide proactive insights for risk mitigation. The predictive component of the system utilizes a machine learning model trained on a diverse dataset of historical accident data, considering factors like as road geometry. The model learns to recognize patterns and correlations between these factors and the occurrence of accidents. As a result, it can predict the probability of an accident happening in real-time based on the ongoing video feed.

The user interface provides a straightforward and intuitive platform for users to upload road CCTV videos and receive instantaneous feedback regarding the likelihood of an accident. The system gives the output of the video by input video as either accident occurs or not with a corresponding confidence score. Users can thus take proactive measures to mitigate potential risks and enhance overall road safety.

The proposed Intelligent Accident Prediction System aims to contribute to the reduction of road accidents by providing a proactive and data-driven approach to identifying potential risk factors. By integrating cutting-edge computer vision technologies, this system represents a step towards the development of smarter, safer, and more efficient transportation systems. The project concludes with the deployment of the trained model for accident detection on external videos. Users can input video, and the model processes and identifies potential accidents

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Road accidents remain a significant global concern, leading to loss of life, injuries, and substantial economic costs. The ability to detect accidents quickly and accurately can save lives, reduce traffic disruptions, and mitigate the severity of the incident. Traditional methods of road accident detection, which often rely on manual observation or delayed reporting, can lead to delayed responses, exacerbating the consequences of accidents. In recent years, advancements in computer vision and machine learning have opened new possibilities for real-time monitoring and prediction of road accidents.

This project proposes an Intelligent Accident Detection and Prediction System, leveraging state-of-the-art computer vision techniques for real-time object detection and machine learning models for accident prediction. By analyzing road surveillance videos through advanced video processing, the system can detect ongoing accidents and assess the probability of an accident happening based on real-time road conditions. The system incorporates historical accident data and factors such as road geometry, traffic flow, and environmental conditions, allowing it to predict accidents proactively.

1.2 OBJECTIVE

The primary goal of this system is to provide an efficient and timely method of detecting accidents, thereby enabling faster responses and enhancing road safety. The user-friendly interface allows users to upload road CCTV videos, and the system quickly analyzes the footage, providing feedback on whether an accident is detected, along with a confidence score that reflects the system's certainty. By offering real-time insights, this system aids in reducing the risks associated with road accidents and contributes to the development of safer transportation systems.

CHAPTER 2

SYSTEM STUDY

2.1 EXISTING SYSTEM

In existing system delays in manual prediction and response may result in missed opportunities to implement preventive measures and mitigate the severity of accidents. Accidents that could have been prevented with timely responses may lead to unnecessary traffic congestion, affecting the overall flow of transportation systems.Manual prediction may not provide timely information to emergency services, hindering their ability to respond promptly to accidents and provide assistance.In cases where accidents could lead to road closures or require evacuation, delays in prediction may limit the time available for effective traffic control and management.

2.1.1 DISADVANTAGE:

- This method requires manpower.
- Waste of time for manual process.
- Low efficiency
- Automatic monitoring is impossible.
- Limiting the accuracy of predictions.

2.2 PROPOSED SYSTEM:

This proposed system can analyze data in real-time, allowing for immediate identification of potential risk factors and timely accident predictions. This system utilizing advanced data mining algorithms and machine learning models can enhance the accuracy of accident predictions compared to manual methods. With timely and accurate predictions, authorities can implement preventive measures proactively, reducing the likelihood and severity of accidents. Timely predictions enable authorities to identify areas with a high risk of accidents, allowing for targeted infrastructure improvements and safety enhancements.

2.2.1 ADVANTAGE:

- It removes human errors that commonly occur during manual analysis.
- User friendliness and minimum time required.
- High Efficiency.
- Simple technique used.

2.3 SOFTWARE AND HARDWARE REQUIREMENTS

HARDWARE SPECIFICATION:

Processor	**:**	Intel icore 7 5^{th} gen
Hard disk	**:**	500 GB
Ram	**:**	12 GB
Keyboard	**:**	Logitech of 104 keys
Mouse	**:**	Logitech mouse
Monitor	**:**	14 inch samtron monitor

SOFTWARE SPECIFICATION:

Front-end	**;**	Python
Language	**:**	Python
Operation system	**:**	Windows 10
Tools	**:**	Python IDLE

SOFTWARE FEATURES

Python:

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

Python is free:

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes. A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, MAC OS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

Python is portable:

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.).

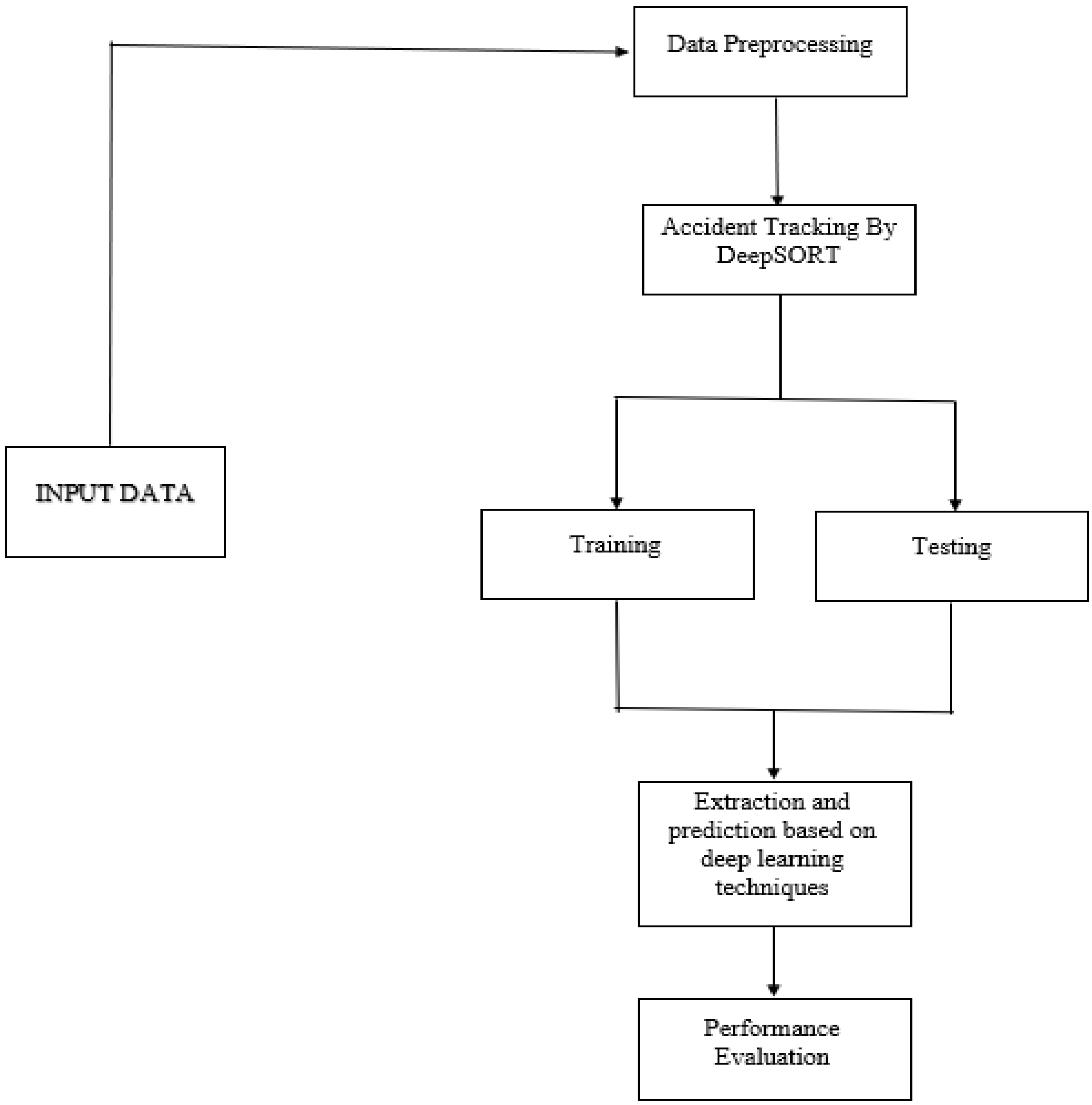
Python is simple:

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way. A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

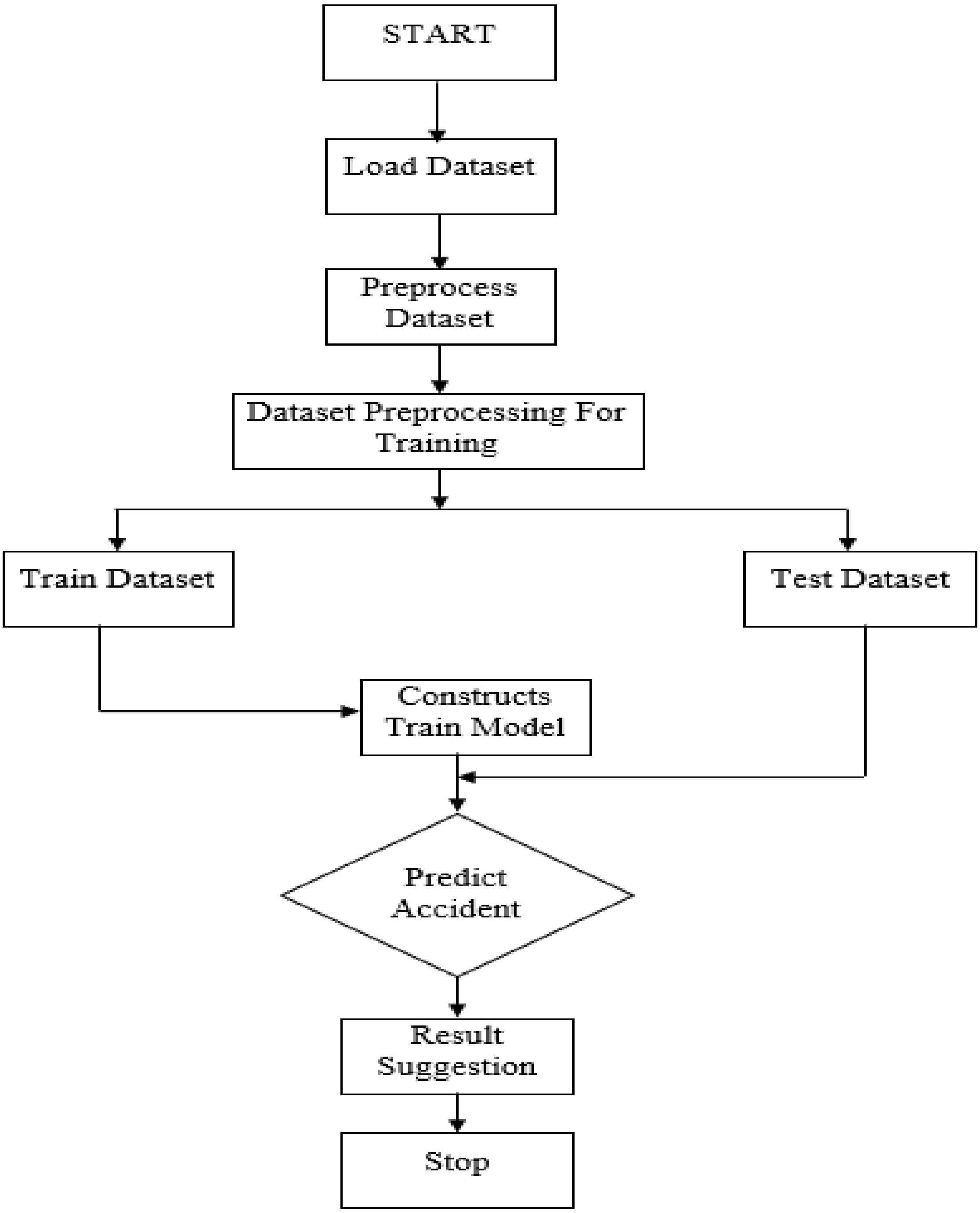
CHAPTER 3

SYSTEM DESIGN

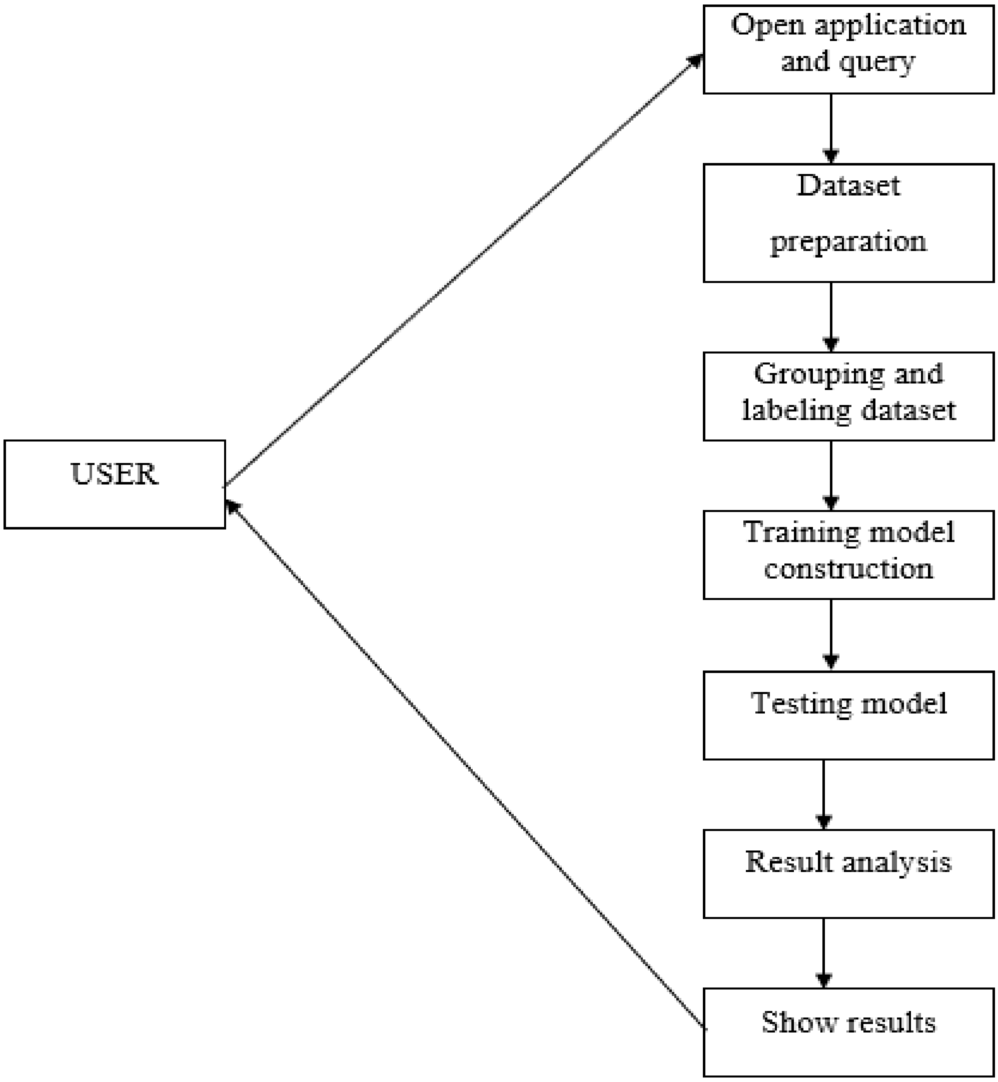
3.1 SYSTEM ARCHITECTURE DIAGRAM



3.2 SYSTEM FLOWCHART



3.3 USE CASE DIAGRAM



3.4 INPUT AND OUTPUT DESIGN

INPUT DESIGN

The input design ensures a seamless and efficient process for users to upload road surveillance videos. The system accepts video files in various formats such as MP4, AVI, and MOV. The interface includes an upload button where users can select and submit a video for processing. The uploaded video undergoes frame-by-frame analysis using computer vision algorithms. Key parameters such as vehicle movement, collision detection, and environmental conditions are extracted. The model then evaluates these factors to determine accident probabilities in real time.

OUTPUT DESIGN

The output design focuses on presenting accident detection results in an intuitive and actionable manner. Once a video is processed, the system provides a clear indication of whether an accident has occurred. The result is displayed as “Accident Detected” or “No Accident” along with a confidence score. The system may also highlight critical frames where an accident is detected for further review. Additionally, users receive real-time alerts if an accident is predicted, allowing for quick intervention. The design ensures that output data is easily interpretable and facilitates prompt decision-making.

3.5 MODULE & DESCRIPTION

MODULES:

- Data Acquisition Module
- Image Preprocessing Module
- Object Detection And Tracking Module
- Density Estimation Module
- Accident Flow Analysis Module:
- User Interface Module
- Integration Module

3.5.1 Data Acquisition Module:

Capture real-time accident scenarios or utilize pre-recorded accident videos for analysis. Use video capturing libraries in Python (e.g., OpenCV) to fetch input accident data.

3.5.2 Image Preprocessing Module:

Enhance the quality of input video frames, correct for lighting conditions, and filter out noise. Apply image processing techniques, including contrast adjustment, histogram equalization, and noise reduction, to improve the accuracy of subsequent computer vision algorithms.

3.5.3 Object Detection and Tracking Module:

Detect and track vehicles within the video frames to obtain real-time accident prediction. Utilize deep learning models for accident detection and tracking. Employ algorithms to associate detected objects across consecutive frames.

3.5.4 Accident Flow Analysis Module:

Analyze the current accident flow patterns to identify congestion and bottlenecks. Utilize the obtained density data to assess the state of accident.

3.5.5 User Interface Module:

Provide a user-friendly interface for system monitoring and control. Design a graphical user interface (GUI) using Python GUI frameworks to display real-time accident probability, and a user-friendly dashboard.

CHAPTER 4

SOFTWARE TESTING

4.1 FUNCTIONAL TESTING

Functional testing is essential to ensure the system performs its intended tasks effectively. This includes verifying that the accident detection model accurately identifies accidents from video footage, such as vehicle collisions or sudden stops. Additionally, the accident prediction model is tested to assess its ability to forecast accident-prone conditions based on real-time factors like road conditions and traffic density. Furthermore, the real-time feedback feature is tested to ensure that the system provides immediate and accurate information to the user after video analysis.

- **Test Case:** Verify that the system correctly detects accidents, predicts accident-prone conditions, and provides real-time feedback after video analysis.
- **Expected Output:** The system accurately identifies accidents, forecasts risks based on real-time factors, and delivers timely and precise feedback to users.

4.2 PERFORMANCE TESTING

Performance testing is conducted to assess the speed and efficiency of the system, particularly in processing video feeds in real-time. The system is tested to ensure it can process long-duration videos and handle high-resolution inputs with minimal delays, providing immediate feedback to users. Scalability is also tested to check whether the system can handle multiple video uploads at once, and load testing is performed to ensure that the system can function efficiently even under high usage conditions.

- **Test Case:** Verify that the system processes long and high-resolution videos quickly, supports multiple uploads, and maintains performance under heavy load.
- **Expected Output:** The system efficiently handles video processing with minimal delay, scales well with multiple uploads, and remains stable under high traffic.

4.3 ACCURACY TESTING

The accuracy of both the accident detection and accident prediction models is a key focus of testing. The detection accuracy is evaluated using metrics like precision, recall, and F1-score, which measure the model's ability to identify true positives (correct accidents), true negatives (correctly ignoring non-accidents), and its overall effectiveness.

- **Test Case:** Verify that the accident detection model correctly identifies accidents and that the prediction model accurately forecasts accident-prone conditions.
- **Expected Output:** The detection model achieves high precision, recall, and F1-score, while the prediction model provides accurate forecasts with minimal error.

4.4 USABILITY TESTING

Usability testing ensures that the system provides an intuitive and user-friendly experience for traffic authorities or other end-users. This testing involves evaluating the user interface to ensure that it is easy to navigate and that users can quickly upload videos and interpret the results. Feedback is collected from test users to identify areas of improvement in the design of the interface, ensuring that the system delivers a seamless experience, even for non-technical users.

- **Test Case:** Verify that users can easily navigate the interface, upload videos, and interpret results without confusion.
- **Expected Output**: Users should be able to move smoothly between sections, upload videos without difficulty, and understand results with clear, well-structured data.

4.5 STRESS AND EDGE CASE TESTING

Stress testing is performed to evaluate how the system handles extreme scenarios. The system is subjected to edge cases such as poor-quality video inputs, extreme lighting conditions, or unusual road conditions (e.g., dense fog or rain) to test its robustness in challenging environments. The objective is to ensure that the system can still detect accidents and predict risks effectively, even when facing unforeseen conditions.

- **Test Case:** Verify system performance with low-resolution videos, poor lighting, and adverse weather conditions.
- **Expected Output:** The system should still function with acceptable accuracy, providing reliable accident detection and risk predictions despite challenging inputs.

4.6 SECURITY TESTING

Security is a critical aspect of system testing to protect user data and ensure that the system is safe from unauthorized access. The system undergoes data encryption testing to ensure that video uploads and other sensitive information are securely transmitted. Additionally, access control mechanisms are evaluated to ensure that only authorized users can upload videos or access the system's features, preventing misuse or data breaches.

- **Test Case:** Verify that video uploads and sensitive data are encrypted and that only authorized users can access system features.
- **Expected Output:** Data remains secure during transmission, unauthorized users are restricted, and access control mechanisms prevent security breaches.

4.7 REGRESSION TESTING

Regression testing is conducted after any updates or changes are made to the system to ensure that new modifications do not introduce new errors or cause any previously functioning features to break. This ensures that the system continues to perform correctly and reliably after improvements or bug fixes.

- **Test Case:** Verify that all core functionalities, including video upload, processing, and result interpretation, work correctly after updates.
- **Expected Output:** The system performs as expected without new errors, and previously working features remain intact.

4.8 USER ACCEPTANCE TESTING (UAT)

User Acceptance Testing (UAT) is carried out to determine whether the system meets the needs and expectations of the end-users. The system is tested in real-world conditions by actual traffic authorities or safety personnel who provide feedback on its effectiveness, accuracy, and usability. Based on their feedback, further adjustments may be made to refine the system's performance.

- **Test Case:** Verify that traffic authorities can efficiently use the system for video uploads, accident detection, and risk analysis in real-world conditions.
- **Expected Output:** Users find the system effective, accurate, and easy to use, with any necessary refinements implemented based on feedback.

CHAPTER 5

CONCLUSION & FUTURE ENHANCEMENT

5.1. CONCLUSION

The Intelligent Accident Detection and Prediction System represents a significant advancement in road safety technology, combining cutting-edge machine learning, computer vision, and real-time data processing to enhance the ability to detect and predict accidents. By leveraging the power of object detection algorithms and predictive models, the system is capable of accurately identifying accidents in real-time and predicting accident-prone conditions based on various factors such as road geometry, traffic density, and weather conditions. The system provides immediate, actionable insights to traffic authorities, enabling them to take preventive actions and improve road safety.

The implementation of this system has demonstrated the effectiveness of using video surveillance, historical accident data, and real-time environmental inputs to create a proactive approach to accident prevention. The integration of real-time feedback into a user-friendly interface further enhances its accessibility, allowing non-technical users to easily interact with the system and respond promptly to detected risks. By automating accident detection and prediction, the system helps to minimize human error, enhance decision-making, and improve overall safety on the roads.

5.2. FUTURE ENHANCEMENT

While the system provides a robust solution for accident detection and prediction, there are several opportunities for future enhancement that could further elevate its performance and utility. One potential improvement is the incorporation of deep learning techniques such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory Networks (LSTMs) to enhance the system's ability to predict accidents based on temporal patterns in traffic behavior. This would allow the system to consider the flow of traffic over time, improving the accuracy of accident predictions.

Another future enhancement could be the integration of additional data sources to improve prediction accuracy. For instance, incorporating vehicle telemetry data such as speed, braking, and acceleration can provide deeper insights into potential accident scenarios. Additionally, integrating weather forecasting systems and real-time road condition reports would allow the system to make more precise predictions based on environmental variables, further enhancing its reliability in predicting accidents under different weather and road conditions.

The expansion of the system's capabilities to detect pedestrian accidents, bicycle accidents, and other non-motorized incidents would be another valuable enhancement. Training the system to detect these types of accidents would increase its applicability, making it useful in a wider range of road safety applications.

Moreover, the system could benefit from edge computing, allowing for faster processing and reducing the latency associated with cloud-based solutions. This would enable real-time accident detection with even lower response times, especially in environments where immediate actions are critical, such as smart cities or high-traffic zones.

Lastly, user engagement and customization features can be enhanced by allowing traffic authorities or road safety agencies to customize risk thresholds, adjust the sensitivity of predictions, or receive notifications about specific accident-prone areas. Providing more detailed, actionable reports based on detected patterns could also enhance the decision-making process for infrastructure planning and resource allocation.

In conclusion, while the Intelligent Accident Detection and Prediction System has already made a considerable impact in improving road safety, ongoing enhancements and advancements in technology will continue to refine and expand its capabilities, ensuring its long-term effectiveness and contribution to safer roads worldwide.

CHAPTER 6

BIBLIOGRAPPHY

(i) REFERENCE

1. **Zhang, X., & Zhang, L.** (2018). A Deep Learning Approach to Road Accident Prediction. Journal of Transportation Safety & Security, 10(4), 232-245.
2. **Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.** (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.
3. **Chen, X., & Cheng, Y.** (2020). Predicting Traffic Accident Risk Using Machine Learning. Transportation Research Part C: Emerging Technologies, 112, 263-275.
4. **Roth, F., & Jansen, S.** (2017). Integrating Smart Traffic Surveillance Systems for Accident Detection. IEEE Intelligent Transportation Systems Magazine, 9(3), 45-56.
5. **Almeida, R., & Pinto, J.** (2019). Machine Learning for Accident Prediction and Prevention: A Survey. Journal of Intelligent Transportation Systems, 23(6), 1-16.
6. **He, K., Zhang, X., Ren, S., & Sun, J.** (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.
7. **Kang, C., & Zhang, X.** (2018). Traffic Accident Detection Using Computer Vision and Machine Learning. Journal of Computer Vision and Image Understanding, 167, 1-11.
8. **Kumar, P., & Sahu, P.** (2021). Smart Traffic Monitoring System Using Convolutional Neural Networks for Road Accident Detection. *International* Journal of Engineering Research & Technology, 10(7), 789-800.
9. **Zhou, X., & Han, J.** (2020). Intelligent Traffic Systems and Safety: A Review of the Latest Developments in Predictive Modeling. Transportation Research Part A: Policy and Practice, 130, 88-102.
10. **Saha, S., & Hossain, M.** (2020). Real-Time Traffic Accident Prediction Using Machine Learning. Journal of Traffic and Transportation Engineering, 7(1), 35-45.

(ii) WESITES

Python:

1. https://www.python.org/
2. https://docs.python.org/
3. https://realpython.com/
4. https://www.geeksforgeeks.org/python-programming-language/
5. https://www.w3schools.com/python/

Anaconda Prompt:

1. https://www.anaconda.com/
2. https://docs.anaconda.com/anaconda/
3. https://docs.conda.io/en/latest/
4. https://www.datacamp.com/tutorial/conda-python-tutorial
5. https://www.geeksforgeeks.org/anaconda-navigator-and-anaconda-prompt/

CHAPTER 7

APPENDIX

(i) SAMPLE CODE

```
from camera import startapplication

startapplication()

"cells": [
 {
  "cell_type": "code",
  "execution_count": 1,
  "id": "c22e4849",
  "metadata": {
   "_cell_guid": "b1076dfc-b9ad-4769-8c92-a6c4dae69d19",
   "_uuid": "8f2839f25d086af736a60e9eeb907d3b93b6e0e5",
   "execution": {
    "iopub.execute_input": "2022-06-06T11:22:04.284474Z",
    "iopub.status.busy": "2022-06-06T11:22:04.283842Z",
    "iopub.status.idle": "2022-06-06T11:22:10.971949Z",
    "shell.execute_reply": "2022-06-06T11:22:10.971144Z",
    "shell.execute_reply.started": "2022-06-06T10:46:28.537996Z"
   },
   "papermill": {
    "duration": 6.715324,
    "end_time": "2022-06-06T11:22:10.972115",
    "exception": false,
```

```
    "start_time": "2022-06-06T11:22:04.256791",
    "status": "completed"
  },
   "tags": []
  },
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "\n"
    ]
   }
  ],
  "source": [
   "import numpy as np\n",
   "import pandas as pd \n",
   "import matplotlib.pyplot as plt\n",
   "import tensorflow as tf\n",
   "from tensorflow.keras import layers\n",
   "from time import perf_counter \n",
   "import os\n",
   "from keras.callbacks import ModelCheckpoint\n",
   "from keras.models import load_model\n",
```

```
    "from tensorflow.keras.utils import plot_model"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "75070279",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2022-06-06T11:22:11.006494Z",
     "iopub.status.busy": "2022-06-06T11:22:11.005804Z",
     "iopub.status.idle": "2022-06-06T11:22:11.008733Z",
     "shell.execute_reply": "2022-06-06T11:22:11.008225Z",
     "shell.execute_reply.started": "2022-06-06T10:46:31.286079Z"
    },
    "papermill": {
     "duration": 0.021664,
     "end_time": "2022-06-06T11:22:11.008874",
     "exception": false,
     "start_time": "2022-06-06T11:22:10.987210",
     "status": "completed"
    },
    "tags": []
   },
```

"outputs": [],
"source": [
"## Defining batch specfications\n",
"batch_size = 100\n",
"img_height = 250\n",
"img_width = 250"
]
},
{
"cell_type": "code",
"execution_count": 3,
"id": "eed3d696",
"metadata": {
"execution": {
"iopub.execute_input": "2022-06-06T11:22:11.043329Z",
"iopub.status.busy": "2022-06-06T11:22:11.042666Z",
"iopub.status.idle": "2022-06-06T11:22:11.281141Z",
"shell.execute_reply": "2022-06-06T11:22:11.281663Z",
"shell.execute_reply.started": "2022-06-06T10:46:31.292379Z"
},
"papermill": {
"duration": 0.258275,
"end_time": "2022-06-06T11:22:11.281841",
"exception": false,

"start_time": "2022-06-06T11:22:11.023566",
"status": "completed"
},
"tags": []
},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Found 791 files belonging to 2 classes.\n"
]
}
],
"source": [
"## loading training set\n",
"training_data = tf.keras.preprocessing.image_dataset_from_directory(\n",
"   'data/train',\n",
"   seed=42,\n",
"   image_size= (img_height, img_width),\n",
"   batch_size=batch_size,\n",
"   color_mode='rgb'\n",
")"
]

},
{
"cell_type": "code",
"execution_count": 4,
"id": "d52bb780",
"metadata": {
"execution": {
"iopub.execute_input": "2022-06-06T11:22:11.318440Z",
"iopub.status.busy": "2022-06-06T11:22:11.317741Z",
"iopub.status.idle": "2022-06-06T11:22:11.437737Z",
"shell.execute_reply": "2022-06-06T11:22:11.438464Z",
"shell.execute_reply.started": "2022-06-06T10:46:31.868234Z"
},
"papermill": {
"duration": 0.141084,
"end_time": "2022-06-06T11:22:11.438707",
"exception": false,
"start_time": "2022-06-06T11:22:11.297623",
"status": "completed"
},
"tags": []
},
"outputs": [
{

```
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "Found 98 files belonging to 2 classes.\n"
   ]
  }
 ],
 "source": [
  "## loading validation dataset\n",
  "validation_data =  tf.keras.preprocessing.image_dataset_from_directory(\n",
  "    'data/val',\n",
  "    seed=42,\n",
  "    image_size= (img_height, img_width),\n",
  "    batch_size=batch_size,\n",
  "    color_mode='rgb'\n",
  ")"
 ]
},
{
 "cell_type": "code",
 "execution_count": 5,
 "id": "a5216ed0",
 "metadata": {
  "execution": {
```

"iopub.execute_input": "2022-06-06T11:22:11.474428Z",
"iopub.status.busy": "2022-06-06T11:22:11.473769Z",
"iopub.status.idle": "2022-06-06T11:22:11.597522Z",
"shell.execute_reply": "2022-06-06T11:22:11.596930Z",
"shell.execute_reply.started": "2022-06-06T10:46:31.998384Z"
},
"papermill": {
"duration": 0.142358,
"end_time": "2022-06-06T11:22:11.597671",
"exception": false,
"start_time": "2022-06-06T11:22:11.455313",
"status": "completed"
},
"tags": []
},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Found 100 files belonging to 2 classes.\n"
]
}
],

```
"source": [
 "## loading testing dataset\n",
 "testing_data = tf.keras.preprocessing.image_dataset_from_directory(\n",
 "    'data/test',\n",
 "    seed=42,\n",
 "    image_size= (img_height, img_width),\n",
 "    batch_size=batch_size,\n",
 "    color_mode='rgb'\n",
 ")"
]
},
{
"cell_type": "code",
"execution_count": 6,
"id": "0f6cb045",
"metadata": {
 "execution": {
  "iopub.execute_input": "2022-06-06T11:22:11.639051Z",
  "iopub.status.busy": "2022-06-06T11:22:11.638119Z",
  "iopub.status.idle": "2022-06-06T11:22:11.642875Z",
  "shell.execute_reply": "2022-06-06T11:22:11.642232Z",
  "shell.execute_reply.started": "2022-06-06T10:46:32.123352Z"
 },
 "papermill": {
```

"duration": 0.029358,

"end_time": "2022-06-06T11:22:11.643040",

"exception": false,

"start_time": "2022-06-06T11:22:11.613682",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"data": {

"text/plain": [

"<_PrefetchDataset element_spec=(TensorSpec(shape=(None, 250, 250, 3), dtype=tf.float32, name=None), TensorSpec(shape=(None,), dtype=tf.int32, name=None))>"

]

},

"execution_count": 6,

"metadata": {},

"output_type": "execute_result"

}

],

"source": [

"testing_data"

]

},

{
"cell_type": "code",
"execution_count": 7,
"id": "cca4892e",
"metadata": {
"execution": {
"iopub.execute_input": "2022-06-06T11:22:11.682246Z",
"iopub.status.busy": "2022-06-06T11:22:11.681351Z",
"iopub.status.idle": "2022-06-06T11:22:11.685497Z",
"shell.execute_reply": "2022-06-06T11:22:11.684898Z",
"shell.execute_reply.started": "2022-06-06T10:46:32.135442Z"
},
"papermill": {
"duration": 0.025718,
"end_time": "2022-06-06T11:22:11.685637",
"exception": false,
"start_time": "2022-06-06T11:22:11.659919",
"status": "completed"
},
"tags": []
},
"outputs": [
{
"data": {

"text/plain": [
"['Accident', 'Non Accident']"
]
},
"execution_count": 7,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"class_names = training_data.class_names\n",
"class_names"
]
},
{
"cell_type": "code",
"execution_count": 8,
"id": "5176cc78",
"metadata": {
"execution": {
"iopub.execute_input": "2022-06-06T11:22:11.724575Z",
"iopub.status.busy": "2022-06-06T11:22:11.723900Z",
"iopub.status.idle": "2022-06-06T11:22:11.729144Z",
"shell.execute_reply": "2022-06-06T11:22:11.729655Z",

"shell.execute_reply.started": "2022-06-06T10:46:32.148210Z"
},
"papermill": {
"duration": 0.027549,
"end_time": "2022-06-06T11:22:11.729835",
"exception": false,
"start_time": "2022-06-06T11:22:11.702286",
"status": "completed"
},
"tags": []
},
"outputs": [],
"source": [
"## Configuring dataset for performance\n",
"AUTOTUNE = tf.data.experimental.AUTOTUNE\n",
"training_data = training_data.cache().prefetch(buffer_size=AUTOTUNE)\n",
"testing_data = testing_data.cache().prefetch(buffer_size=AUTOTUNE)"
]
},
{
"cell_type": "code",
"execution_count": 9,
"id": "82bc89e3",
"metadata": {

"execution": {
"iopub.execute_input": "2022-06-06T11:22:11.768660Z",
"iopub.status.busy": "2022-06-06T11:22:11.767963Z",
"iopub.status.idle": "2022-06-06T11:22:11.826933Z",
"shell.execute_reply": "2022-06-06T11:22:11.826211Z",
"shell.execute_reply.started": "2022-06-06T10:46:32.163777Z"
},
"papermill": {
"duration": 0.079717,
"end_time": "2022-06-06T11:22:11.827076",
"exception": false,
"start_time": "2022-06-06T11:22:11.747359",
"status": "completed"
},
"tags": []
},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"\n"
]
}

],
"source": [
"## Defining Cnn\n",
"model = tf.keras.models.Sequential([\n",
" layers.BatchNormalization(),\n",
" layers.Conv2D(32, 3, activation='relu'), # Conv2D(f_size, filter_size, activation) # relu, sigmoid, softmax\n",
" layers.MaxPooling2D(), # MaxPooling\n",
" layers.Conv2D(64, 3, activation='relu'),\n",
" layers.MaxPooling2D(),\n",
" layers.Conv2D(128, 3, activation='relu'),\n",
" layers.MaxPooling2D(),\n",
" layers.Conv2D(256, 3, activation='relu'),\n",
" layers.MaxPooling2D(),\n",
" layers.Flatten(),\n",
" layers.Dense(512, activation='relu'),\n",
" layers.Dense(len(class_names), activation= 'softmax')\n",
"])\n",
"\n",
"model.compile(optimizer='adam',loss='sparse_categorical_crossentropy', metrics=['accuracy'])"
]
},
{
"cell_type": "code",

"execution_count": 10,

"id": "3c69caf5",

"metadata": {

"execution": {

"iopub.execute_input": "2022-06-06T11:22:11.866838Z",

"iopub.status.busy": "2022-06-06T11:22:11.866008Z",

"iopub.status.idle": "2022-06-06T11:22:12.150115Z",

"shell.execute_reply": "2022-06-06T11:22:12.149113Z",

"shell.execute_reply.started": "2022-06-06T10:46:33.004393Z"

},

"papermill": {

"duration": 0.306291,

"end_time": "2022-06-06T11:22:12.150305",

"exception": false,

"start_time": "2022-06-06T11:22:11.844014",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"name": "stdout",

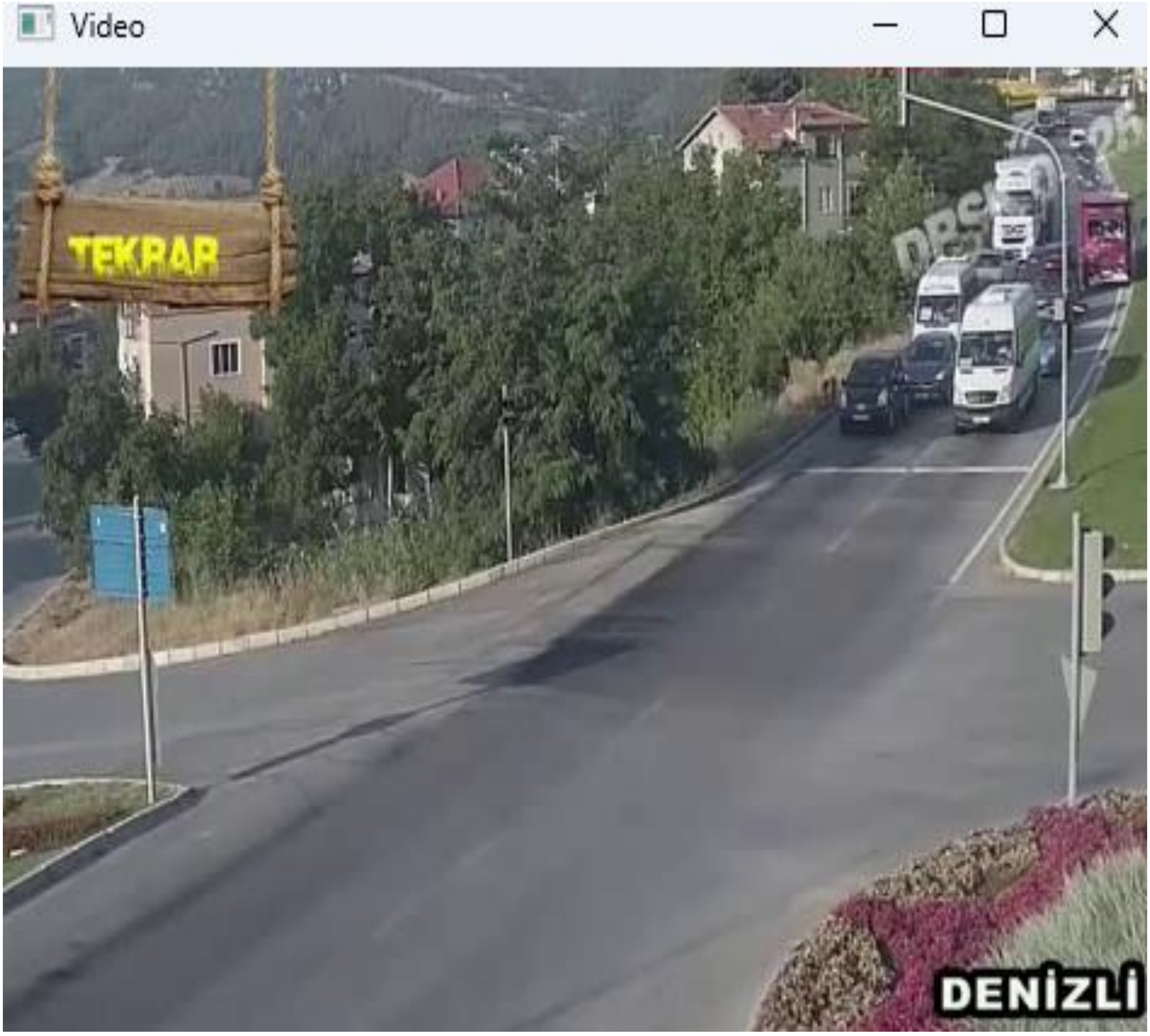
"output_type": "stream",

"text": [

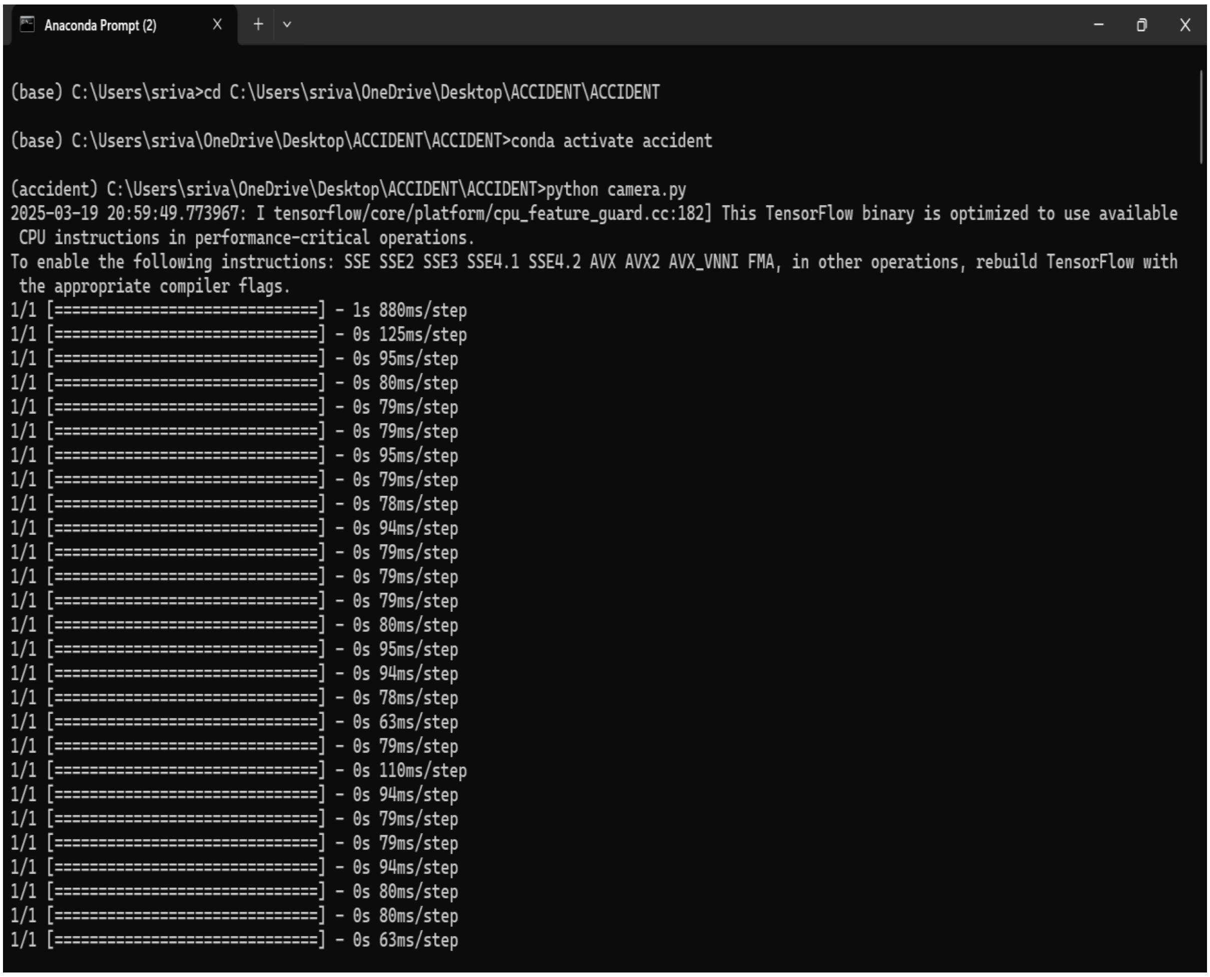
```
    "Model: \"sequential\"\n",
from keras.models import model_from_json
import numpy as np


class AccidentDetectionModel(object):


  class_nums = ['Accident', "No Accident"]


  def __init__(self, model_json_file, model_weights_file):
    # load model from JSON file
    with open(model_json_file, "r") as json_file:
      loaded_model_json = json_file.read()
      self.loaded_model = model_from_json(loaded_model_json)


    # load weights into the new model
    self.loaded_model.load_weights(model_weights_file)
    self.loaded_model.make_predict_function()


  def predict_accident(self, img):
    self.preds = self.loaded_model.predict(img)
    return AccidentDetectionModel.class_nums[np.argmax(self.preds)], self.preds
```

(ii) SCREENSHOTS

Video:

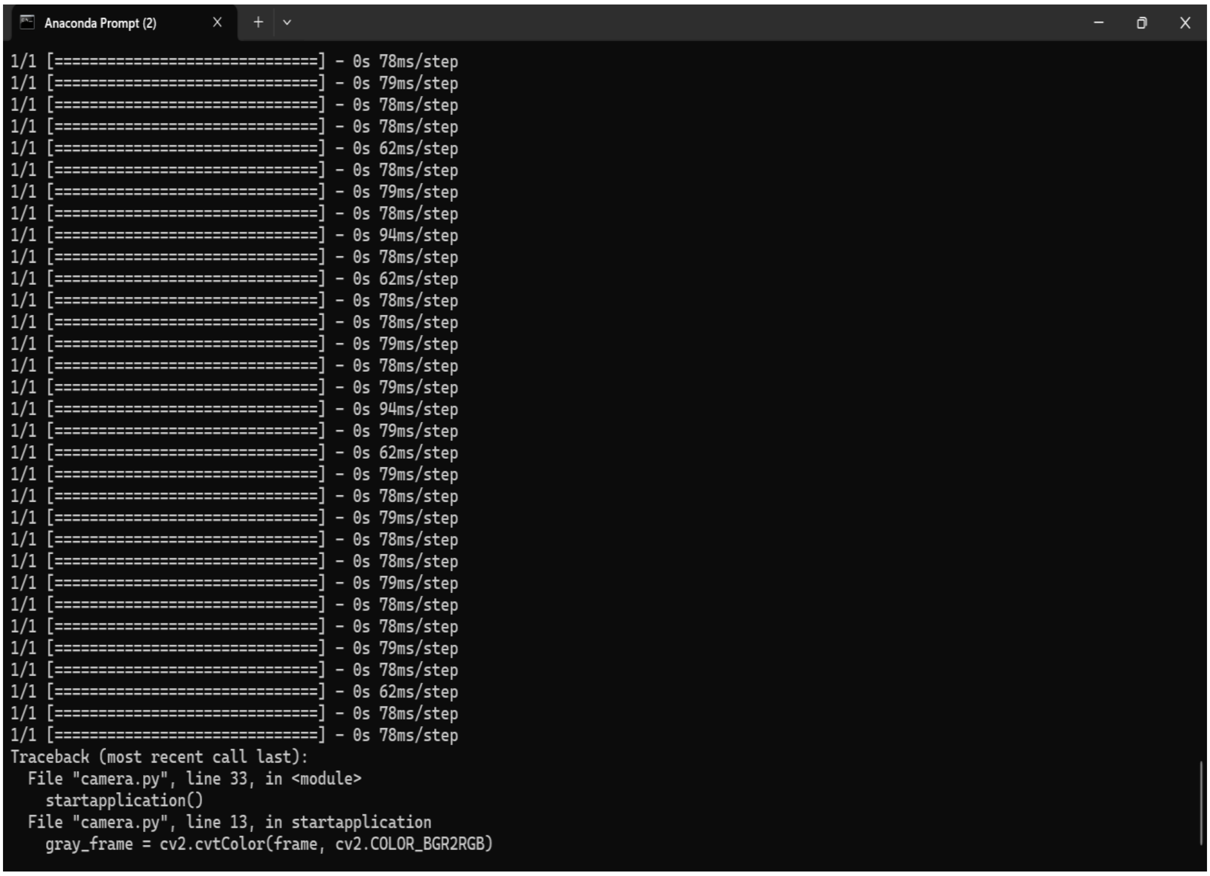


Started To Detecting:

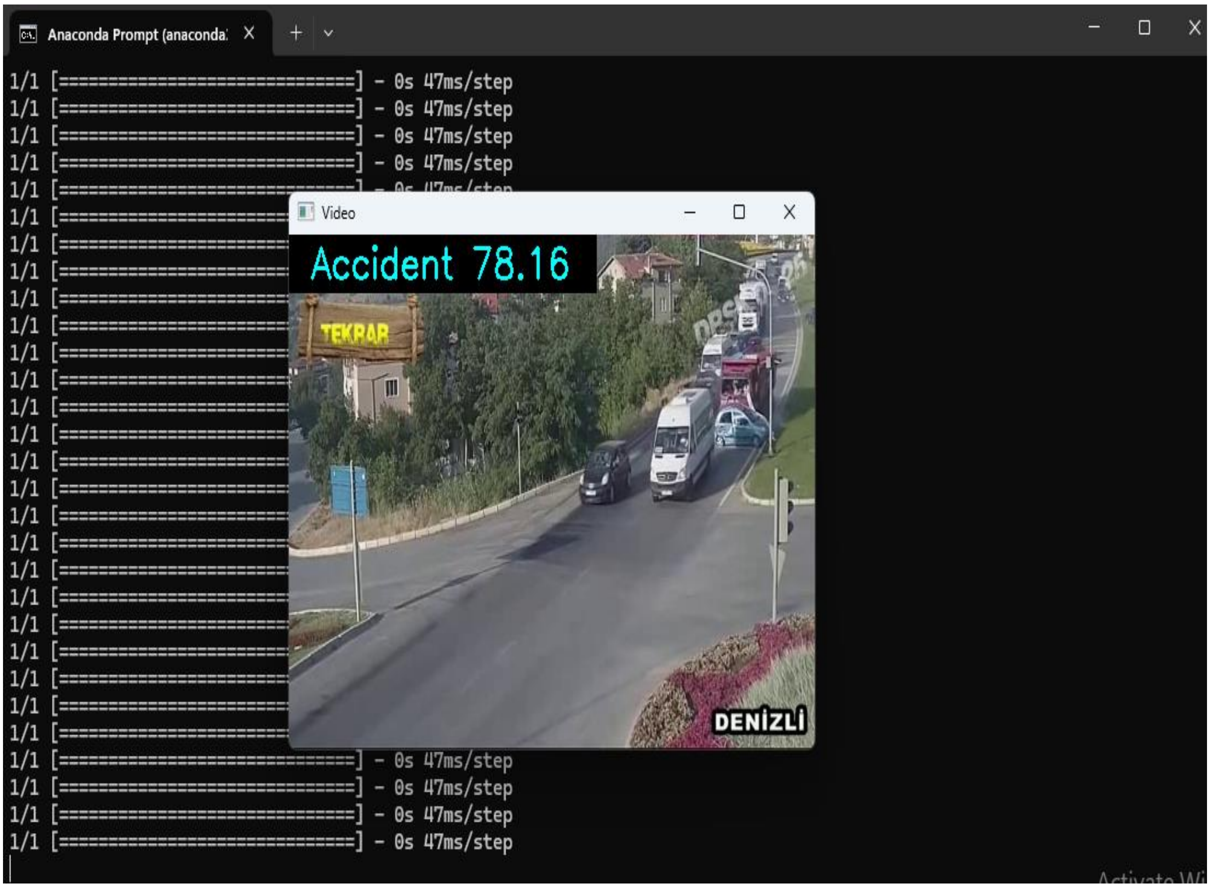


```
(base) C:\Users\sriva>cd C:\Users\sriva\OneDrive\Desktop\ACCIDENT\ACCIDENT

(base) C:\Users\sriva\OneDrive\Desktop\ACCIDENT\ACCIDENT>conda activate accident

(accident) C:\Users\sriva\OneDrive\Desktop\ACCIDENT\ACCIDENT>python camera.py
2025-03-19 20:59:49.773967: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available
 CPU instructions in performance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with
 the appropriate compiler flags.
1/1 [==============================] - 1s 880ms/step
1/1 [==============================] - 0s 125ms/step
1/1 [==============================] - 0s 95ms/step
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 95ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 94ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 95ms/step
1/1 [==============================] - 0s 94ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 63ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 110ms/step
1/1 [==============================] - 0s 94ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 94ms/step
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 63ms/step
```

Accident Detected:



Detected:


Anaconda Prompt (anaconda
1/1 [==============================] - 0s 47ms/step
Video
Accident 78.16
TEKRAR
DENİZLİ