VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Santhibastawad Road, Machhe
Belagavi - 590018, Karnataka, India

PROJECT WORK PHASE- 2 (18CSP83) REPORT

ON

"DESIGN OF INTELLIGENT SURVEILLANCE SYSTEM FOR MULTIPLE ANOMALY DETECTION USING DEEP LEARNING"

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

BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING

For the Academic Year 2023-2024

Submitted by

SATHVIK S UPADHYA	**1JS20IS092**
SREERANGA J	**1JS20IS104**
TANMAYA R	**1JS20IS110**
VISHAL K T	**1JS20IS118**

Under the Guidance of
Dr. ABHIJITH H V
Assoc Professor Dept. of ISE, JSSATE

2023-2024

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060

JSS MAHAVIDYAPEETHA, MYSURU

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Project Work Phase - 2 (18CSP83) Report entitled "DESIGN OF INTELLIGENT SURVEILLANCE SYSTEM FOR MULTIPLE ANOMALY DETECTION USING DEEP LEARNING" is a bonafide work carried out by SATHVIK S UPADHYA [1JS20IS092], SREERANGA J [1JS20IS104], TANMAYA R [1JS20IS110], VISHAL K T [1JS20IS118] in partial fulfillment for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University Belagavi during the year 2023- 2024.

Signature of the Guide	Signature of the HOD
Dr ABHIJITH H V	**Dr. REKHA P M**
Assoc. Professor	Professor& Head
Dept. of ISE	Dept. of ISE
JSSATE, Bengaluru	JSSATE, Bengaluru

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So with gratitude, we acknowledge all those whose guidance and encouragement crowned my effort with success.

First and foremost, we would like to thank his **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Bhimasen Soragaon**, Principal, JSSATE, Bengaluru for providing an opportunity to carry out the Project Work Phase – 2 (18CSP83) as a part of our curriculum in the partial fulfillment of the degree course.

We express our sincere gratitude for our beloved Head of the department, Dr**. Rekha P M**, for her co-operation and encouragement at all the moments of our approach.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guide **Dr ABHIJITH H V**, Assoc. Professor, for the constant encouragement, valuable help and assistance in every possible way.

We are thankful to the Project Coordinators **Dr. Nagamani N P**, Assoc. Professor and **Mrs. Sahana V** Asst. Professor, for their continuous co-operation and support.

We would like to thank all **ISE department teachers** and **non-teaching staff** for providing us with their valuable guidance and for being there at all stages of our work.

SATHVIK S UPADHYA [1JS20IS092]
SREERANGA J [1JS20IS104]
TANMAYA R [1JS20IS110]
VISHAL K T [1JS20IS118

DESIGN OF INTELLIGENT SURVEILLANCE SYSTEM FOR MULTIPLE ANOMALY DETECTION USING DEEP LEARNING

ABSTRACT

This project focuses on an advanced automated surveillance system designed for real-time anomaly detection in public spaces equipped with CCTV cameras. Specifically, it addresses the critical need for enhanced security and public safety by targeting the detection of fights, car crashes, falls, and Fire Detection. Leveraging computer vision and deep learning techniques, the system continuously monitors video feeds, swiftly identifies anomalies, and promptly notifies security personnel. The project's multifaceted objectives encompass improving security measures, reducing the reliance on manual surveillance, optimizing resource allocation, ensuring cost-effectiveness, and fostering continuous system enhancement. By proactively detecting and responding to security incidents, this project aims to create a safer environment for the public, law enforcement, and security personnel. The anticipated outcomes encompass a notable enhancement in security measures and public safety, a substantial reduction in the manual monitoring burden, cost savings, efficient resource allocation, and a steadfast commitment to system refinement based on historical data analysis and user feedback. Furthermore, this project aspires to contribute to the advancement of technology in computer vision and machine learning, setting a precedent for the adoption of automated surveillance systems in various public settings.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: -

INTRODUCTION

In the contemporary landscape of heightened security concerns, public safety stands as a paramount priority, necessitating proactive measures to safeguard citizens. While many public places are equipped with CCTV camera systems, the conventional approach of manual monitoring for anomalies such as fights, car crashes, falls, and Fire Detection proves to be both resource-intensive and susceptible to errors. In response to this challenge, we present a comprehensive solution in the form of an advanced automated surveillance system.

This innovative system is designed to autonomously monitor CCTV camera feeds through the application of cutting-edge computer vision and deep learning techniques. Its primary focus lies in the real-time detection of critical incidents, including fights, car crashes, falls, and Fire Detection. By harnessing the power of artificial intelligence, the system not only significantly enhances overall security but also alleviates the burden of manual monitoring, ensuring a more efficient and effective response to security incidents.

A key strength of our proposed system lies in its effortless incorporation with prevailing infrastructure commonly found in public spaces. This integration facilitates a smooth transition to an automated surveillance paradigm, maximizing the utility of the pre-existing CCTV camera network. Moreover, the system is meticulously designed to adhere to privacy regulations, striking a delicate balance between enhanced security measures and the protection of individual privacy rights. The inclusion of an intuitive user interface further ensures that security personnel and administrators can easily configure and customize system settings, fostering user engagement and efficient management.

Anticipated outcomes of the project extend beyond immediate security enhancements. They include a notable improvement in public safety, achieved through the system's proactive monitoring capabilities. Efficient resource allocation is a direct consequence, leading to substantial cost savings over time. Furthermore, the project is poised to contribute to advancements in the fields of computer vision and machine learning, setting a precedent for the evolution of automated surveillance systems in public spaces.

1.1 PROBLEM STATEMENT

Majority of public place are equipped with CCTV cameras for 24 x 7 Surveillance. But it requires manual monitoring for Fight Detection, Car Crash Detection, Fall Detection or Fire Detection. Manual Monitoring cannot be in Realtime manner. Therefore, an automated solution capable of real-time anomaly detection is imperative.

1.2 OBJECTIVES

This Project aims to satisfy the following objectives

1) Design of Deep Learning Model for Real Time Detection of Car Crash Detection at Public Place.
2) Design of Deep Learning Model for Real Time Detection of Fight at Public Place.
3) Design of Deep Learning Model for Real Time Detection of Fire Detection at Public Place.
4) Design of Deep Learning Model for Real Time Detection of Fall Detection at Public Place.

1.3 SOLUTION TO THE PROBLEM

To address the challenge of manual monitoring for critical incidents in public spaces, we propose an innovative solution leveraging advanced computer vision and deep learning techniques. Our approach involves collecting a diverse dataset of CCTV footage annotated with instances of fights, car crashes, falls, and fire incidents. Through preprocessing and augmentation, we ensure data quality and model generalization. We then train a deep learning model, such as YOLOv7, on this dataset, optimizing performance metrics through validation and testing. The integration of this trained model into existing CCTV infrastructure enables real-time incident detection, with an intuitive user interface facilitating efficient system management while adhering to privacy regulations.

This comprehensive solution not only enhances public safety by proactively detecting incidents but also streamlines resource allocation and reduces manual monitoring burdens. By seamlessly integrating with existing infrastructure and prioritizing privacy, our system represents a significant advancement in automated surveillance for public spaces. Furthermore, its iterative enhancement process guarantees adaptability to evolving security requirements, thus contributing to the continuous progress of computer vision and machine learning technologies in the domain of public safety

1.4 EXISTING TECHNIQUE

The current approach to surveillance in public spaces predominantly relies on conventional CCTV camera systems, which necessitate manual monitoring for the detection of critical incidents such as fights, car crashes, falls, and fire outbreaks. However, this manual monitoring process is resource-intensive, prone to errors, and often lacks real-time responsiveness. Security personnel tasked with monitoring numerous camera feeds face challenges in promptly identifying and responding to incidents as they occur. Moreover, the reliance on human observers introduces the potential for oversight and fatigue, further compromising the effectiveness of surveillance efforts. Overall, the existing technique of manual monitoring falls short in providing timely and accurate incident detection, highlighting the need for a more advanced and automated surveillance solution.

1.5 PROPOSED TECHNIQUE

The proposed technique introduces an advanced automated surveillance system designed to address the limitations of manual monitoring in public spaces. Leveraging cutting-edge computer vision and deep learning techniques, our solution aims to autonomously analyze CCTV camera feeds in real-time for the detection of critical incidents such as fights, car crashes, falls, and fire outbreaks. By leveraging artificial intelligence capabilities, the system can effectively detect anomalies and initiate alerts without requiring continuous human oversight.

Key to our approach is the integration of a robust deep learning model, such as YOLOv7, trained on a diverse dataset of annotated CCTV footage. This model is optimized to accurately detect and classify incidents while minimizing false positives. Moreover, our solution emphasizes the integration with current infrastructure to ensure compatibility with pre-existing CCTV camera networks commonly present in public spaces. Furthermore, privacy concerns are addressed through meticulous design, striking a balance between enhanced security measures and individual privacy rights.

1.6 SCOPE OF THE PROJECT

Data Collection and Preparation

Gathering a diverse dataset of annotated CCTV footage containing instances of fights, car crashes, falls, and fire incidents. Preprocessing and augmenting the dataset to ensure data quality and model generalization.

Model Development and Training

Selecting an appropriate deep learning architecture, such as YOLOv7, and training it on the annotated dataset to accurately detect and classify incidents in real-time. Refining the model to enhance performance metrics such as accuracy, precision, recall, and F1 score.

Integration and Deployment

Developing an intuitive interface for system management and integrating the trained model into existing CCTV infrastructure. Implementing real-time incident detection capabilities and alert mechanisms to notify security personnel promptly.

Evaluation and Improvement

Assessing the system's performance in real-world settings through validation and testing. Collecting feedback from users to identify areas for improvement and iterating on the model and system architecture accordingly.

1.7 ORGANISATION OF REPORT

1) Chapter 2 contains a Literature Survey on Different Surveillance Systems, Manual and Automated, Deep Learning Anomaly Detection Models which uses CNN or RNN, YOLO models
2) Chapter 3: System Requirements Specification
3) Chapter 4: Gives the complete System Design, it explains Flow chart, System Architecture, Use Case Diagram and Sequence Diagram.
4) Chapter 5: Gives the details about the implantation details about our project including
5) the architecture.
6) Chapter 6: It focuses on different forms test on the completed project such as unit testing, Integration testing, System Testing.
7) Chapter 7: Gives the conclusion and points out the future enhancements.
8) Chapter 8: It has the snapshots of the application developed application.

Chapter 2: -

LITERATURE SURVEY

In recent years, the field of automated surveillance and anomaly detection has witnessed significant growth, driven by advancements in computer vision and deep learning techniques. Researchers have explored a diverse range of methodologies aimed at enhancing surveillance systems' effectiveness, with a particular emphasis on identifying anomalies within video streams. Innovative approaches utilizing deep learning models, such as CNNs and RNNs, have shown promise in recognizing complex anomalies like abnormal behaviors and object movements. However, challenges persist, including issues with noise handling and delays in anomaly detection.

The integration of intelligent video surveillance systems (IVSS) has introduced comprehensive architectures equipped with IP cameras and advanced algorithms for abnormal activity detection. These systems leverage techniques like SVM and deep learning frameworks to safeguard assets from threats like vandalism and theft. Despite their robustness, challenges remain in feature extraction and classification algorithms, leading to potential misclassifications, especially in dynamic scenarios.

Our project seeks to build upon existing research by implementing a cutting-edge automated surveillance system that harnesses the latest advancements in computer vision and deep learning. By leveraging techniques such as the YOLOv7 object detection model and advanced anomaly detection algorithms, our system offers enhanced accuracy and real-time anomaly detection capabilities. Additionally, our architecture prioritizes user-friendly interfaces and seamless integration with external tools like PowerBI dashboards, providing security personnel with comprehensive insights and actionable intelligence. Through rigorous testing and optimization, we aim to overcome the limitations identified in previous studies, ensuring a robust and reliable surveillance solution for public safety and security.

Chapter 3: -

SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

3.1.1 IDE USED – VS CODE

Visual Studio Code (VS Code), developed by Microsoft, stands as a highly favored source code editor renowned for its lightweight design, extensive customization options, and efficiency in supporting modern programming languages and workflows. Here's a succinct overview of VS Code:

Features and Functionality

VS Code boasts a rich array of features aimed at enhancing the coding experience. It encompasses support for syntax highlighting, code completion, navigation, and debugging functionalities.

Integrated Development Environment (IDE) Features

Despite being primarily a code editor, VS Code incorporates several IDE-like features to streamline development processes. These include integrated terminal support, built-in Git version control, debugging tools, task automation, and more, providing a comprehensive development environment.

Fig 3.1 Visual Code Logo

3.1.2 PYTHON

The project's primary language, Python, serves as the cornerstone for deep learning model development. Renowned for its extensive libraries and flexibility, Python provides an ideal environment for implementing sophisticated algorithms associated with anomaly detection.

General-Purpose Language

Python is a versatile programming language suitable for a multitude of applications, spanning from web development and data analysis to scientific computing, artificial intelligence, machine learning, and automation, among others

Large Ecosystem of Libraries and Frameworks

Python boasts a vibrant ecosystem of third-party libraries and frameworks like Django and Flask are popular frameworks for web development, NumPy and pandas for data analysis, TensorFlow and PyTorch for machine learning, and matplotlib and seaborn for data visualization.

Dynamic Typing and Strong Typing

Python's dynamic typing feature implies that variable types are determined at runtime based on the assigned value. Additionally, Python is strongly typed, meaning that it enforces strict type-checking rules to prevent unintended type conversions and ensure code reliability.

Fig 3.2 Python Logo

3.1.3 DEEP LEARNING LIBRARIES

1.OpenCV

Object Detection and Tracking Central to the system's capabilities, OpenCV is utilized for object detection and tracking purposes. Leveraging its robust computer vision functionalities, OpenCV extracts vital information from video streams, enabling the identification and monitoring of anomalies. Its real-time image processing capabilities align with the project's emphasis on timely anomaly detection in public spaces.

Image and Video Processing

OpenCV offers an extensive collection of tools and functions for processing images and videos It boasts many image processing operations, including basic manipulations (e.g., resizing, cropping, rotating), filtering (e.g., blurring, sharpening, edge detection), color space conversion, and feature extraction.

Computer Vision Algorithms

OpenCV includes many computer vision algorithms for object detection, feature detection and matching, object tracking, optical flow estimation, and camera calibration. These algorithms enable developers to build sophisticated applications for tasks like facial recognition, object recognition, gesture recognition, and augmented reality.

Fig 3.3 OpenCV Logo

2.Keras

Keras is an open-source deep learning framework written in Python which has a high-level API for building and training neural network models. It is user-friendly, modular, and extensible, making it ideal for deep learning practitioners.

Support for Multiple Backends

Keras supports multiple backends, including TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK). This backend-agnostic design allows developers to seamlessly switch between different backends without changing their code, providing flexibility and portability across different environments.

Versatility and Flexibility

Keras supports a wide range of neural network architectures and applications, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GANs), and more. It also provides support for both traditional supervised learning tasks (classification, regression) and advanced techniques such as reinforcement learning and transfer learning.

Integration with TensorFlow

While Keras can be used as a standalone library, it is tightly integrated with TensorFlow, the most popular deep learning framework. Keras is included as part of the TensorFlow library (tf. Keras), making it the default high-level API for building and training neural network models in TensorFlow.

Fig 3.4 Keras Logo

3.PyTorch

PyTorch is an open-source machine learning framework developed by Facebook's AI Research lab (FAIR). It provides a dynamic computational graph mechanism, making it particularly well-suited for deep learning research and experimentation.

Dynamic Computational Graph

PyTorch uses a dynamic computational graph approach, where computational graphs are constructed dynamically as operations are executed. This allows for more flexibility and expressiveness in model construction and enables easier debugging and experimentation compared to static computational graph frameworks.

Autograd Mechanism

PyTorch includes an automatic differentiation mechanism called Autograd, which automatically computes gradients of tensors with respect to variables in the computational graph. This enables efficient gradient-based optimization algorithms such as stochastic gradient descent (SGD) and backpropagation.

CUDA Support

PyTorch provides seamless integration with NVIDIA's CUDA platform for GPU acceleration, allowing for efficient parallel computation on GPUs. This CUDA support enables PyTorch to leverage the computational power of modern GPUs for accelerating training and inference tasks.

Fig 3.5 PyTorch Logo

3.1.4 POSTGRE SQL

PostgreSQL is a powerful, open-source relational database management system (RDBMS) known for its robustness, reliability, and advanced features. Here's an overview of PostgreSQL:

Relational Database Management System

PostgreSQL is a relational database management system based on the SQL (Structured Query Language) standard. It provides a structured approach to storing and managing data in tables with relationships defined by keys and constraints.

ACID Compliance

PostgreSQL adheres to the ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring data integrity and transactional consistency. ACID compliance guarantees that database transactions are processed reliably, even in the event of system failures or crashes.

Advanced Features

PostgreSQL offers a wide range of advanced features and capabilities, including support for complex data types (e.g., JSON, XML, arrays), full-text search, geospatial data processing, and custom data types and functions. It also supports advanced SQL features such as common table expressions (CTEs), window functions, and recursive queries.

Fig 3.6 PostgreSQL Logo

3.1.5 FRONT END TECHNOLOGIES

HTML, CSS, JavaScript On the front end, HTML, CSS, and JavaScript collectively contribute to creating a visually appealing and responsive user interface. HTML structures the content, CSS styles the layout, and JavaScript adds interactivity. This combination enhances the user experience, providing a seamless interaction with the system and contributing to the creation of an intuitive dashboard for monitoring and configuring the surveillance system.

1.HTML

HTML (Hypertext Markup Language) serves as the backbone of the web, providing the structure and semantics for web pages. Here's a concise overview of HTML:

Markup Language

HTML is a markup language used to create the structure of web pages by defining elements and their attributes. It utilizes tags to enclose content and provide instructions to web browsers on how to display the content.

Semantic Structure

HTML emphasizes semantic markup, allowing developers to define the purpose and meaning of each element within a web page. This semantic structure not only improves accessibility but also enhances search engine optimization (SEO) by providing meaningful context to search engines.

Element Hierarchy

HTML documents are structured hierarchically using elements such as <html>, <head>, <body>, and various other tags to organize content. Elements can be nested within each other to create complex layouts and structures.

Continuous Evolution

HTML is constantly evolving to meet the demands of modern web development. New features and enhancements are regularly introduced through updates and new specifications, ensuring that HTML remains relevant and adaptable to emerging technologies and trends.

Fig 3.7 HTML Logo

2. CSS

CSS (Cascading Style Sheets) is a fundamental technology used for styling and formatting web pages, enhancing their visual presentation and layout. Here's a concise overview of CSS:

Styling and Formatting

CSS enables developers to define the appearance of HTML elements by specifying properties such as colour, font, size, spacing, and alignment. By separating content from presentation, CSS facilitates consistent styling across multiple web pages and ensures a cohesive user experience.

Selectors and Declarations

CSS employs selectors to target specific HTML elements and apply styling rules to them. Selectors can target elements based on their type, class, ID, attributes, or relationship to other elements. Styling rules, known as declarations, consist of property-value pairs that define the desired appearance of selected elements.

Media Queries

With media queries, developers can apply different styles based on the characteristics of the device or viewport, such as screen width, height, orientation, and resolution.

Animations and Transitions

CSS supports animations and transitions, allowing developers to create visually appealing effects such as fades, slides, rotations, and transformations. Keyframe animations enable precise control over the animation timeline and interpolation between different states of an element.

Fig 3.8 CSS Logo

3.JAVA SCRIPT

JavaScript, often abbreviated as JS, is a dynamic programming language used primarily for client-side web development. Here's a concise overview of JavaScript:

Client-Side Scripting

JavaScript is primarily executed on the client side, meaning it runs in the user's web browser. It enables interactive features and dynamic behaviour within web pages, enhancing user experience and interactivity.

Core Functionality

JavaScript allows developers to manipulate the content of web pages in real-time, enabling actions such as modifying HTML elements, handling user input through forms, and dynamically updating page content without requiring a page refresh.

DOM Manipulation

JavaScript has access to the Document Object Model (DOM), a programming interface which shows the structure of HTML documents as a hierarchical tree of objects. Developers can use JavaScript to traverse and manipulate the DOM, dynamically modifying elements, attributes, and styles to update the appearance and behaviour of web pages.

Asynchronous Programming

JavaScript supports asynchronous programming patterns through features such as callbacks, promises, and async/await syntax. This allows developers to execute tasks asynchronously, such as fetching data from servers, processing user input, and performing animations, without blocking the main execution thread.

Fig 3.9 JavaScript Logo

3.1.6 BACK-END TECHNOLOGIES

1.YOLO

YOLO (You Only Look Once) is a state-of-the-art object detection algorithm known for its speed and accuracy. YOLO divides the input image into a grid and predicts bounding boxes and class probabilities for objects within each grid cell. YOLO v7 is an improved version of the YOLO algorithm, incorporating enhancements and optimizations to further improve detection performance. Here's an overview of YOLO and its version 7:

YOLOv7 Improvements

YOLO v7 builds upon the strengths of previous versions while introducing enhancements to further improve detection performance. These improvements may include architectural changes, optimization techniques, and training strategies aimed at reducing false positives, improving localization accuracy, and increasing detection speed.

Single Pass Inference

YOLO is able to perform object detection in a single pass through the network. This makes YOLO extremely fast in processing images in real-time on standard hardware, making it suitable for applications such as video surveillance, autonomous driving, and robotics.

YOLOv7 Improvements

YOLO v7 builds upon the strengths of previous versions while introducing enhancements to further improve detection performance. These improvements may include architectural changes, optimization techniques, and training strategies aimed at reducing false positives, improving localization accuracy, and increasing detection speed.

Real-World Applications

YOLO and its variants, including YOLO v7, have been widely adopted in real-world applications, including object detection in surveillance systems, object tracking in video streams, and human pose estimation in sports analytics.

Fig 3.10 YOLO v7 Logo

2.Flask Web Framework

Building the Web Interface Facilitating the construction of an interactive web interface, Flask, a lightweight web framework for Python, is employed. Flask's simplicity and extensibility make it a suitable choice for seamlessly integrating various system components.

Microframework

Flask is referred to as a "microframework" because it prioritizes simplicity and minimalism. It provides the essential components needed for web development without imposing unnecessary dependencies or restrictions.

Routing and URL Mapping

Flask allows developers to define URL routes and associate them with specific view functions, which help to handle incoming requests and generating responses. Route decorators are used to specify URL patterns and HTTP methods, making it easy to create RESTful APIs and web applications with clean and intuitive URL structures.

HTTP Request and Response Handling

Flask provides intuitive APIs for working with HTTP requests and responses. Request objects encapsulate incoming HTTP requests, allowing developers to access request data, such as form data, query parameters, and request headers. Response objects enable the generation of HTTP responses with custom status codes, headers, and content.

Fig 3.11 Flask Logo

3.2 HARDWARE REQUIREMENTS

- **High Performance GPU**

  Boosting Machine Learning Efficiency: Integrating high-performance GPUs, such as NVIDIA GeForce RTX 3080 or AMD Radeon RX 6900 XT, is crucial for enhancing the efficiency of machine learning tasks, especially in speech emotion recognition. These GPUs provide accelerated parallel processing capabilities, optimizing the training and execution of complex models.

- **Processor**

  A powerful multi-core processor, such as Intel Core i7-11700K or AMD Ryzen 7 5800X, is essential for handling the computational workload associated with deep learning tasks and real-time video processing. The processor's high clock speed and multi-threading capabilities contribute to faster data processing and model training, enhancing the overall performance of the surveillance system.

- **RAM**

  Adequate RAM capacity is crucial for storing and manipulating large datasets during model training and inference. A minimum of 16 GB of high-speed DDR4 RAM, such as Corsair Vengeance LPX or G.SKILL Ripjaws V Series, is recommended to ensure smooth operation of deep learning tasks and real-time video processing.

- **Storage**

  High-speed storage drives, such as Samsung 970 EVO Plus NVMe SSD or Western Digital Black SN850 NVMe SSD, are essential for fast data access and retrieval. A minimum of 500 GB of SSD storage is recommended for storing datasets, model checkpoints, and system logs.

3.3 FUNCTIONAL REQUIREMENTS

- **Object Detection Accuracy**

  Leveraging the YOLO (You Only Look Once) algorithm, the system must exhibit a high level of accuracy in recognizing and classifying anomalies.

- **Real-time Processing**

  A critical capability of the system is its ability to perform anomaly detection in real-time or near-real-time. Timely response to security incidents is imperative.

- **Multiple Anomaly Types**

  The system should exhibit versatility by being able to detect various types of anomalies. This includes but is not limited to anomalies such as fires, fights, Falls, Car Crashes and abnormal behavior.

3.4 NON-FUNCTIONAL REQUIREMENTS

- **Reliability**

  The system should be highly reliable, with minimal false positives and false negatives in anomaly detection.

- **Performance**

  The system must have high performance with minimal latency to ensure quick anomaly detection and response.

- **Robustness**

  It should be robust against variations in lighting conditions, weather, occlusions, and other environmental factors.

- **Security**

  Ensure that the system is secure against potential attacks, especially in scenarios where the anomaly detection system might be a target for malicious activities.

Chapter 4: -

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The system architecture of the project comprises several interconnected components designed to provide users with seamless access to anomaly detection services. Below is an explanation of the system architecture and its key components

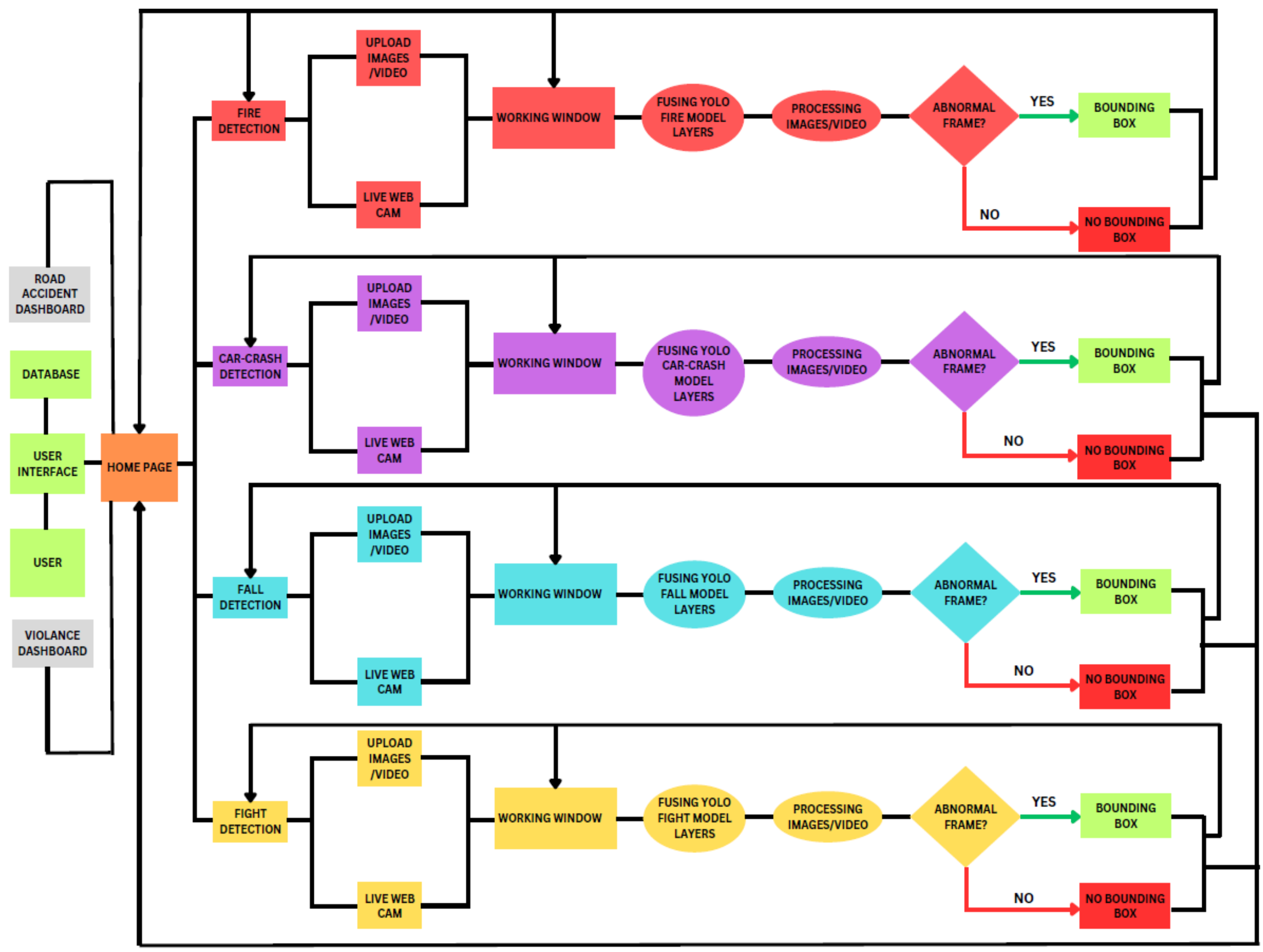


Fig 4.1 System Architecture

- **Database**

  The database component stores user credentials for login and signup functionalities. It contains tables for user authentication and authorization.

- **User Interface (Browser)**

  The user interface is accessible through a web browser, allowing users to interact with the system. It serves as the primary entry point for accessing the system's services.

- **Home Page**

  The home page is the main dashboard for users, providing access to various services and functionalities offered by the system. Users can navigate from the home page to different services, such as fire detection, car crash detection, fall detection, or fight detection.

- **Anomaly Detection Services**

  Each anomaly detection service (fire, car crash, fall, fight) is represented as a separate module within the system architecture. Users can choose to input live video feeds, recorded videos, or images for anomaly detection within each service.

- **Service Modules**

  Within each service module, users have the option to select the type of input (live video, recorded video, or image) they want to analyze for anomalies. Upon selecting an input type, a working window opens, helping users to monitor the processing of the input video or image.

- **Model Integration**

  The system integrates the respective anomaly detection models for fire, car crash, fall, or fight detection. These models consist of layers trained to detect anomalies within input frames.

- **Anomaly Detection Process**

  As the input frames are processed by the respective anomaly detection models, the system identifies abnormal frames. If an anomaly is detected, a bounding box is drawn on the video feed or image, indicating the location of the anomaly detected by the model. If no anomaly is detected, no bounding box is drawn, indicating that the input frame is normal.

- **Interconnectivity**

  Users have the flexibility to switch between live webcam feeds, recorded inputs, or return to the home page from any service module. This interconnectivity ensures a friendly user experience, allowing users to navigate between different functionalities.

- **PowerBI Dashboards**

  Additionally, the home page provides a connection to PowerBI dashboards for visualizing data related to violence in public spaces and road accidents. Users can access these dashboards directly from the home page for data analysis and insights.

4.2 USE CASE DIAGRAM

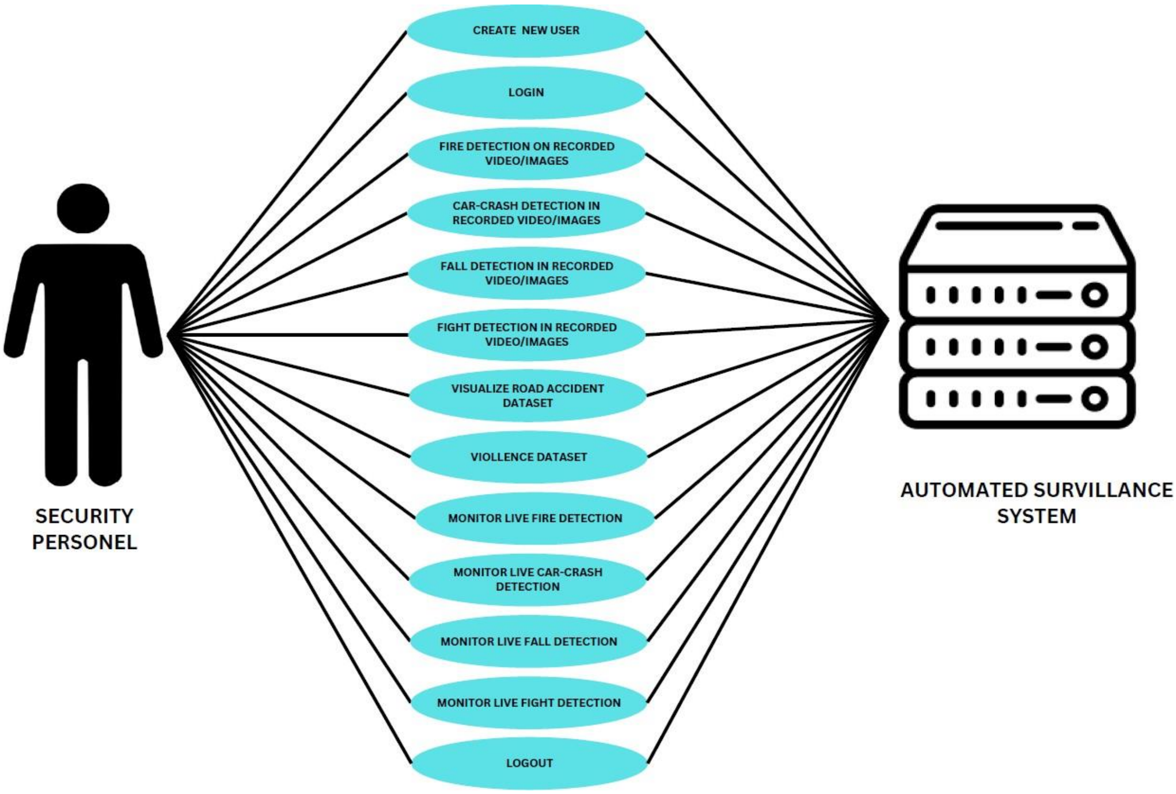


Fig 4.2 Use Case Diagram

Use Case Diagram: Automated Surveillance System

Actor: Security Personnel

Use Cases

1. **Login**

Security personnel logs into the automated surveillance system to access its functionalities.

2. **Create User**

Security personnel creates a new user account in the system for authentication and access control.

3. Logout

Security personnel logs out of the system, ending the current session.

4. Recorded Fire Video/Image

Security personnel selects recorded video or image footage for fire detection analysis.

5. Recorded Car Crash Video/Image

Security personnel selects recorded video or image footage for car crash detection analysis.

6. Recorded Fall Video/Image

Security personnel selects recorded video or image footage for fall detection analysis.

7. Recorded Fight Video/Image

Security personnel selects recorded video or image footage for fight detection analysis.

8. View PowerBI Dashboard - Road Accident

Security personnel views the PowerBI dashboard displaying data and insights on road accidents.

9. View PowerBI Dashboard - Violence in Public Places

Security personnel views the PowerBI dashboard displaying data and insights on violence in public places.

10. Live Fire Detection

Security personnel initiates live fire detection analysis using real-time video feeds.

11. Live Fall Detection

Security personnel initiates live fall detection analysis using real-time video feeds.

12. Live Car Crash Detection

Security personnel initiates live car crash detection analysis using real-time video feeds.

13. Live Fight Detection

Security personnel initiates live fight detection analysis using real-time video feeds.

4.3 SEQUENCE DIAGRAM

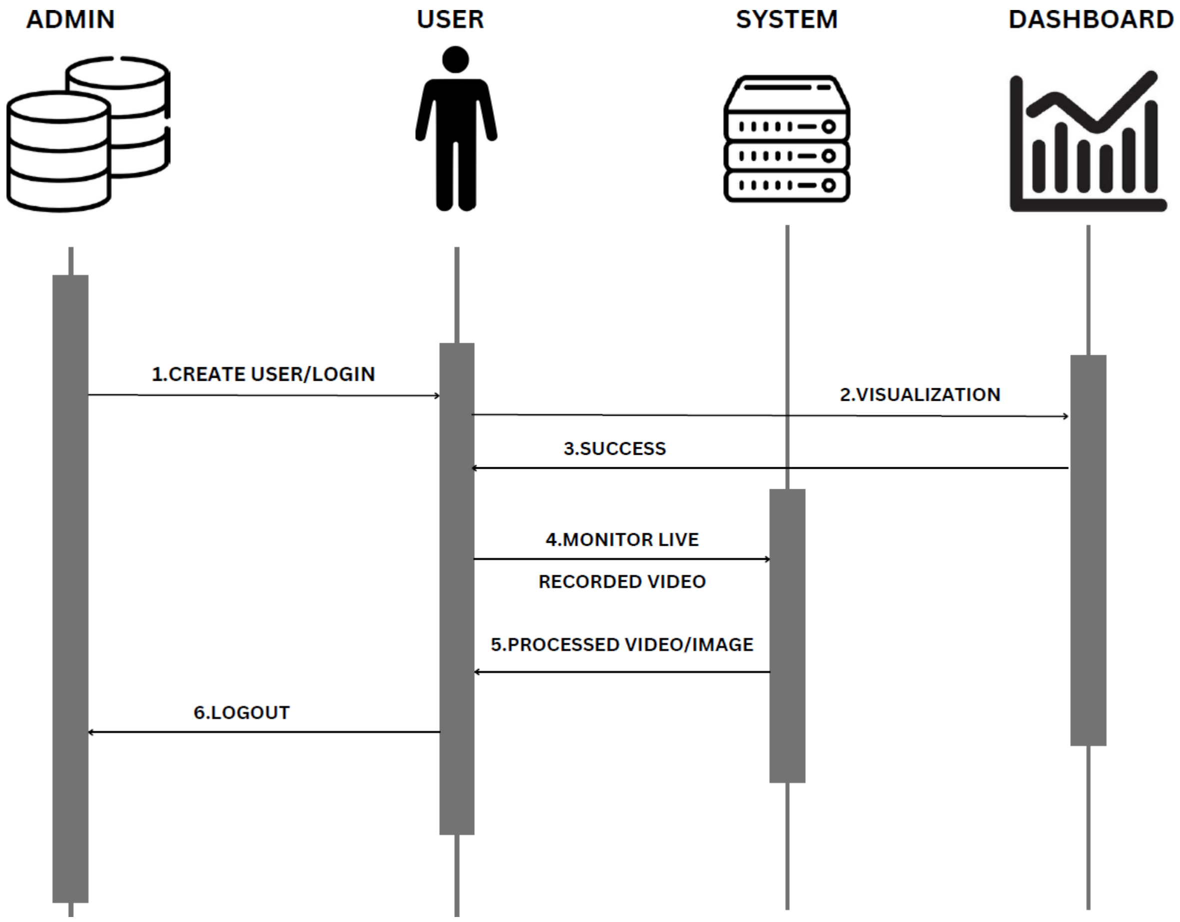


Fig 4.3 Sequence Diagram

1. **Create User/Login**

   AdminUser1 sends a request to the System to create a new user or login. The System processes the request and interacts with the admin component to create a new user account or authenticate the login credentials. The System creates a new user account or verifies the login credentials. The System responds with a success message or authentication failure message to AdminUser1.

2. **Visualization**

   Upon successful login, AdminUser1 requests access to the Dashboard component for data visualization. The System forwards the request to the Dashboard module. Dashboard retrieves relevant data from the admin component and generates visualizations using PowerBI tools. Dashboard sends the data visualizations to AdminUser1 for display.

3. **Success**

   If the login/authentication process is successful, the System notifies AdminUser1 with a success message. AdminUser1 can proceed to access the Dashboard or use the anomaly detection services.

4. **Monitor Live/Recorded Video**

   AdminUser1 requests to monitor live or recorded video for anomaly detection. The System processes the request and interacts with the respective Anomaly Detection Module (fire, car crash, fall, fight). The Anomaly Detection Module analyzes the live or recorded video feed for anomalies. The System receives processed frames from the Anomaly Detection Module and displays them to AdminUser1 for monitoring. This sequence continues as AdminUser1 monitors live or recorded video feeds for anomaly detection.

5. **Processed Video/Image**

   If anomalies are detected in the video frames or images, the System overlays bounding boxes around the abnormal regions. AdminUser1 receives the processed video or images with bounding boxes indicating anomalies. If no anomalies are detected, the System notifies AdminUser1 accordingly

6. **Logout**

   AdminUser1 initiates the logout process by sending a logout request to the System. The System terminates the user session and logs out AdminUser1 from the system.

   AdminUser1 is successfully logged out and the session ends.

4.4 CONTROL FLOW DIAGRAM

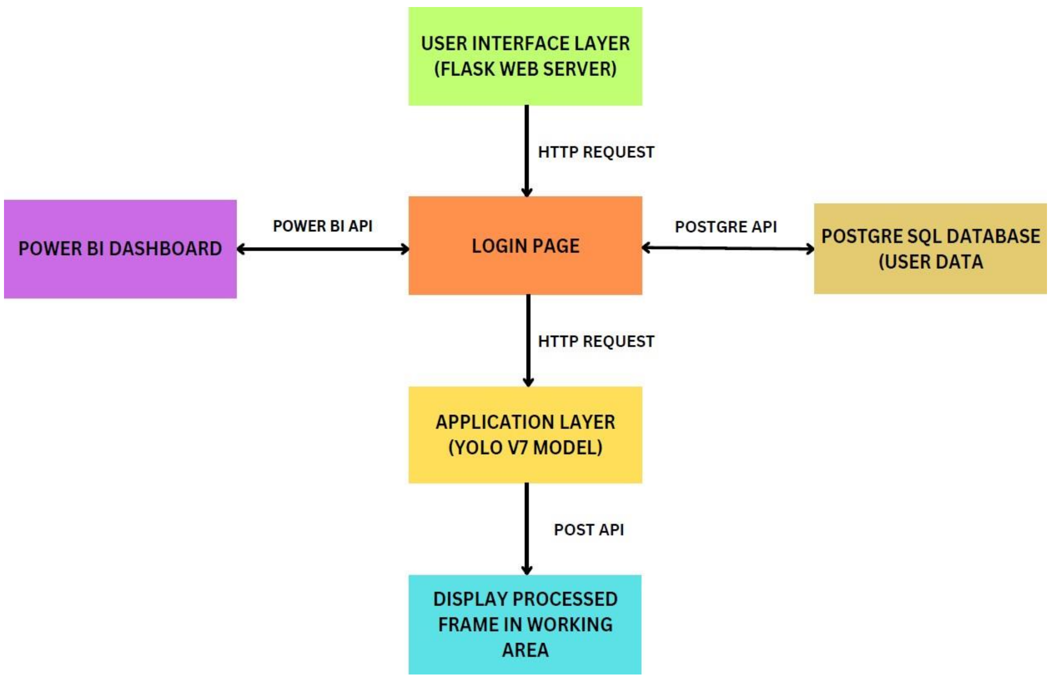


Fig 4.4 Control Flow Diagram

User Interface

The control flow initiates with user interactions through the User Interface layer. Users perform actions such as logging in, creating new accounts, or accessing system features.

Home Page

Upon receiving user interactions, the control flow proceeds to the Home Page. The Home Page serves as a central routing hub for directing requests to different system components.

Postgres Database

If the user initiates actions related to authentication or account management, the control flow directs requests to the Postgres Database. The database verifies user credentials and manages user accounts through a Postgres API.

Application Layer

For surveillance-related tasks, such as anomaly detection, the control flow routes requests to the Application Layer. Here, YOLOv7 models are deployed to process surveillance data and detect anomalies like fire, car crashes, falls, or fights.

Working Area

Processed data from the Application Layer flows to the Working Area for further analysis and visualization. Annotated video feeds or images indicating detected anomalies are presented in the Working Area for user monitoring.

Dashboard

Additionally, the control flow includes access to Power BI dashboards through the Dashboard component. Users can view comprehensive visualizations and analysis of surveillance data related to violence in public spaces and road accidents.

4.5 APIS USED

The automated surveillance system leverages several APIs to enable seamless communication between its components and external services. These APIs facilitate data exchange, authentication, and integration with external platforms, enhancing the system's functionality and interoperability.

1. Postgres API

The Postgres API enables interaction with the Postgres Database, which stores user authentication data and manages user accounts. It provides endpoints for user authentication, account creation, and account management operations such as login, signup, and profile updates. Through the Postgres API, the system verifies user credentials, retrieves user information, and performs database operations securely.

2. YOLOv7 API

The YOLOv7 API serves as the interface for deploying and interacting with YOLOv7 models responsible for anomaly detection in surveillance data. It offers endpoints for processing video feeds or images, detecting anomalies (such as fire, car crashes, falls, or fights), and returning annotated outputs. This API allows the system to integrate advanced computer vision capabilities for real-time anomaly detection and monitoring.

3. Power BI API

The Power BI API facilitates integration with Power BI dashboards, enabling data visualization and analysis of road accidents and violence in public spaces. It provides endpoints for accessing and retrieving data from Power BI dashboards, allowing users to view and analyze relevant statistics and trends. Through the Power BI API, the system enables users to monitor road safety and public security issues efficiently.

4. HTTP API

The HTTP API is used for handling HTTP requests and responses between different system components. It supports communication between the User Interface, Home Page, Application Layer, and Working Area, enabling seamless data flow and control within the system. This API facilitates the exchange of data and commands, ensuring effective coordination and operation of the automated surveillance system.

Chapter 5: -

IMPLEMENTATION

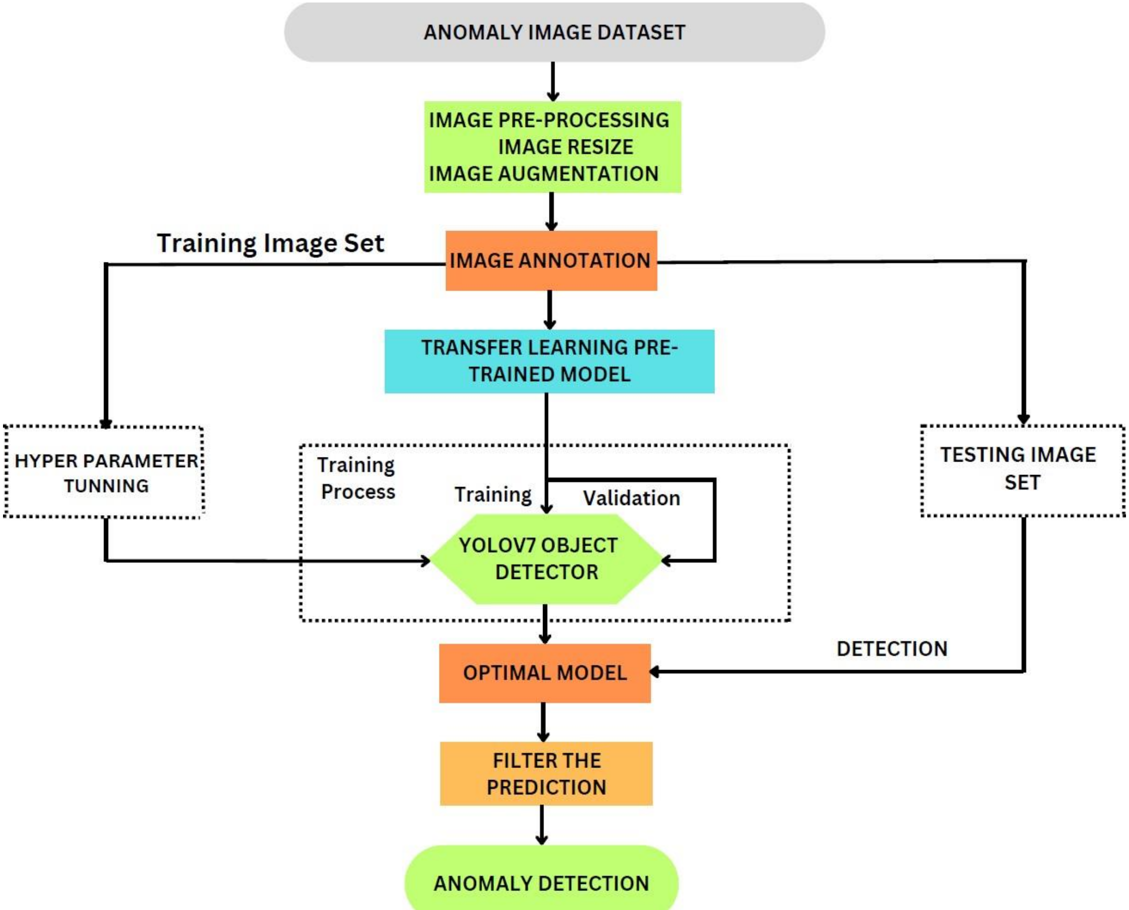


Fig 5.1 Model Implementation

Anomaly Image Data Set

The process begins with the anomaly image data set containing images representing various anomalies such as fire, car crashes, falls, and fights.

Image Pre-processing (Image Resize & Image Augmentation)

The raw images undergo pre-processing steps, which may include tasks like noise reduction, contrast enhancement, and color normalization to prepare them for further analysis. The images are resized to a uniform size and augmented to increase the diversity of the dataset. Augmentation techniques may include rotation, flipping, and cropping.

Image Annotation:

The annotated images are labeled with bounding boxes or masks to indicate the presence and location of anomalies within the images.

Hyperparameter Tuning

On the left branch, hyperparameter tuning is performed to optimize the parameters of the anomaly detection model, such as learning rate, batch size, and optimization algorithm.

Test Image Set

On the right branch, a separate set of test images helps to evaluate the performance of the anomaly detection model.

Transfer Learning Pre-Trained Model

A pre-trained deep learning model, such as YOLOv7, is utilized for anomaly detection. Transfer learning is employed to fine-tune the model on the annotated image dataset.

Training Process

The fine-tuned model is trained on the augmented and annotated image dataset to learn to detect anomalies. The trained YOLOv7 object detector iteratively processes the input images to detect anomalies based on learned patterns and features.

Optimized Model

The model undergoes optimization to improve its performance and efficiency in detecting anomalies.

Detection Process

The optimized model is applied to the test image set to detect anomalies in real-world scenarios.

Filtering the Prediction

Post-processing techniques are applied to filter and refine the predictions generated by the model, reducing false positives and improving detection accuracy.

Anomaly Detection Process

The final process is to detect anomalies within the test images, providing valuable insights for security and safety applications.

5.1 FIRE DETECTION MODEL

- The Fire Detection module utilizes a YOLOv7 PyTorch model trained on pre-processed images to detect fire incidents in real-time surveillance footage.
- The model is trained on a dataset comprising diverse fire scenarios, including indoor and outdoor fires, varying intensities, and different environmental conditions.
- Specific hyperparameters such as learning rate, batch size, and input image size are fine-tuned to optimize the model's performance in detecting fires with high accuracy and efficiency.
- Transfer learning techniques are applied, leveraging features learned from pre-trained models on large-scale datasets, and fine-tuning them to detect fire-related anomalies effectively.
- The model's architecture is tailored to identify common characteristics of fire, such as flame patterns, smoke formations, and temperature variations, enabling timely detection and alerting of fire incidents in surveillance footage.

5.2 CAR CRASH DETECTION MODEL

- The Car Crash Detection module utilizes a YOLOv7 PyTorch model trained on pre-processed images to identify car crashes or collisions in surveillance videos.
- The model is trained on a dataset containing diverse car crash scenarios, including rear-end collisions, side impacts, and rollover accidents.
- Specific hyperparameters such as anchor box sizes, IoU (Intersection over Union) thresholds, and confidence thresholds are optimized to ensure accurate detection of car crashes while minimizing false positives.
- Transfer learning is employed to leverage features learned from pre-trained models on general object detection tasks and adapt them to detect car crash-related anomalies efficiently.

- The model's architecture is designed to recognize distinctive visual cues associated with car crashes, such as sudden changes in velocity, deformation of vehicle structures, and deployment of airbags, enabling prompt detection and response to road traffic accidents.

5.3 FIGHT DETECTION MODEL

- The Fight Detection module utilizes a YOLOv7 PyTorch model trained on pre-processed images to detect physical altercations or fights in surveillance footage.
- The model is trained on a dataset containing diverse fight scenarios, including one-on-one confrontations, group fights, and physical assaults.
- Specific hyperparameters such as object scale, object confidence threshold, and non-maximum suppression threshold are fine-tuned to optimize the model's sensitivity to fight-related activities while reducing false alarms.
- Techniques such as transfer learning are applied to adapt features learned from pre-trained models on general object detection tasks to the specific task of fight detection, enhancing the model's ability to recognize aggressive behaviour and violent interactions accurately.
- The model's architecture is customized to detect visual cues indicative of fights, such as rapid movements, physical contact between individuals, and gestures associated with aggression, facilitating early intervention and mitigation of violent incidents.

5.4 FALL DETECTION MODEL

- The Fall Detection module utilizes a YOLOv7 PyTorch model trained on pre-processed images to identify instances of falls or accidents involving individuals in surveillance videos.
- The model is trained on a dataset comprising diverse fall scenarios, including slips, trips, and accidental falls, captured from various camera viewpoints and environments.

- Specific hyperparameters such as object size, confidence threshold, and post-processing parameters are tuned to optimize the model's ability to detect falls accurately while minimizing false alarms.
- Transfer learning strategies are employed to leverage features learned from pre-trained models on general object detection tasks and adapt them to the task of fall detection, enhancing the model's sensitivity to human body movements indicative of falls.
- The model's architecture is tailored to recognize distinct postures and movements associated with falls, such as sudden changes in orientation, loss of balance, and impacts with the ground, enabling rapid detection and response to potential accidents or medical emergencies.
- These four modules collectively enhance the surveillance system's capabilities to detect and respond to critical incidents, ensuring the safety and security of public spaces effectively.

Chapter 6: -

TESTING

6.1 UNIT TESTING

In Unit Testing individual function/singular behaviors of software are tested and verified. Unit Test is conducted in order to validate every function/module of the software performs as expected. Here, we presented the results of the unit test conducted on each module of the application.

Table 6.1 Unit Test Cases

TEST CASE	OBJECTIVE	ACTUAL OUTPUT	EXPECTED OUTPUT	REMARKS
1	Test whether the contents are displaying properly	Yes, displayed	Yes, displayed	Passed
2	Test whether the application name and logo are displaying properly	Yes, displayed	Yes, displayed	Passed
3	Test whether the menu bar is displaying and working properly	Yes, displayed	Yes, displayed	Passed
4	Test whether the Home page works without the user logging in	Yes, displayed	Yes, displayed	Passed
5	Test Clicking on the home button redirects to the home page	Yes, Redirected	Yes, Redirected	Passed
6	Test Clicking on the services button redirects to the services page	Yes, Redirected	Yes, Redirected	Passed
7	Test Clicking on the team button redirects to the team details page	Yes, Redirected	Yes, Redirected	Passed
8	Test Clicking on the login button redirects to the login page	Yes, Redirected	Yes, Redirected	Passed
9	Test if Services Unlock after logging in	Yes, Unlocked	Yes, Unlocked	Passed
10	Test Dashboard Dropdown Options for road accident and violence detection displayed correctly	Yes, displayed	Yes, displayed	Passed

TEST CASE	OBJECTIVE	ACTUAL OUTPUT	EXPECTED OUTPUT	REMARKS
11	Test After creating a new user, the user is redirected back to the login page	Yes, Redirected	Yes, Redirected	Passed
12	Test Users can only access services after successful login	Yes, Working	Yes, Working	Passed
13	Test Login fails with incorrect username or password	Yes, Working	Yes, Working	Passed
14	Test Login fails if username or password fields are empty	Yes, Working	Yes, Working	Passed
15	Test Attempting to access services without logging in redirects to the login page	Yes, Working	Yes, Working	Passed
16	Test Clicking on a service button redirects to the respective service page	Yes, Redirected	Yes, Redirected	Passed
17	Test The home, about, and services buttons function correctly on the service page	Yes, Working	Yes, Working	Passed
18	Test Users can upload video or image files for anomaly detection	Yes, Working	Yes, Working	Passed
19	Test Uploaded files are processed by the YOLOv7 model for anomaly detection	Yes, Working	Yes, Working	Passed
20	Test Live mode option allows users to stream video for real-time anomaly detection	Yes, Working	Yes, Working	Passed
21	Test Processed image displays the detected anomalies correctly	Yes, Working	Yes, Working	Passed
22	Test Uploaded image or video with fire anomaly detected correctly.	Yes, Fire Detected	Yes, Fire Detected	Passed

TEST CASE	OBJECTIVE	ACTUAL OUTPUT	EXPECTED OUTPUT	REMARKS
23	Test Uploaded image or video with car crash anomaly detected correctly	Yes, Car Crash Detected	Yes, Car Crash Detected	Passed
24	Test Uploaded image or video with fall anomaly detected correctly	Yes, fall Detected	Yes, fall Detected	Passed
25	Test Uploaded image or video with fight anomaly detected correctly	Yes, Fight Detected	Yes, Fight Detected	Passed
26	Test Uploaded image or video without anomalies detected correctly	Yes, Working	Yes, Working	Passed

6.2 INTEGRATION TESTING

Integration testing is conducted to test how individual units will integrate with each other and test them as a group. It is conducted in order to expose the faults in the interaction between the integrated unit components.

Table 6.2 Integration Test Cases

TEST CASE	OBJECTIVE	ACTUAL OUTPUT	EXPECTED OUTPUT	REMARKS
1	Clicking on the services button from the home page redirects to the services page	Yes, Redirected	Yes, Redirected	Passed
2	Clicking on the home button from the services page redirects to the home page	Yes, Redirected	Yes, Redirected	Passed
3	Clicking on the about button from the services page redirects to the about page	Yes, Redirected	Yes, Redirected	Passed
4	Successful login enables access to the services page	Yes, Working	Yes, Working	Passed

TEST CASE	OBJECTIVE	ACTUAL OUTPUT	EXPECTED OUTPUT	REMARKS
5	Attempting to access services without logging in redirects to the login page	Yes, Redirected	Yes, Redirected	Passed
6	Clicking on a service button redirects to the respective service page	Yes, Redirected	Yes, Redirected	Passed
7	The home, about, and services buttons function correctly on the service page	Yes, Working	Yes, Working	Passed
8	Users can upload video or image files for anomaly detection	Yes, Working	Yes, Working	Passed
9	Uploaded files are processed by the YOLOv7 model for anomaly detection	Yes, Working	Yes, Working	Passed
10	Live mode option allows users to stream video for real-time anomaly detection	Yes, Working	Yes, Working	Passed
11	Processed image displays the detected anomalies correctly	Yes, Working	Yes, Working	Passed
12	Fire anomaly detected correctly in integrated system	Yes, Fire Detected	Yes, Fire Detected	Passed
13	Car crash anomaly detected correctly in integrated system	Yes, Car Crash Detected	Yes, Car Crash Detected	Passed
14	Fall anomaly detected correctly in integrated system	Yes, Fall Detected	Yes, Fall Detected	Passed
15	Fight anomaly detected correctly in integrated system	Yes, Fight Detected	Yes, Fight Detected	Passed
16	Test When no anomalies present, they are not detected correctly in integrated system	Yes, Working	Yes, Working	Passed

TEST CASE	OBJECTIVE	ACTUAL OUTPUT	EXPECTED OUTPUT	REMARKS
17	Test New users can be created and login functionality works as expected	Yes, Working	Yes, Working	Passed
18	Select Road Accidents option and Relevant data should be displayed.	Yes, Displayed	Yes, Displayed	Passed
19	Select Violence Detection option and Relevant data should be displayed.	Yes, Displayed	Yes, Displayed	Passed

6.3 SYSTEM TESTING

System testing validates the complete and full software product. System Testing is done to evaluate the end-to-end system specifications. System testing for the automated surveillance system involves a comprehensive evaluation of its various components and functionalities to ensure that it operates effectively in real-world scenarios. This testing encompasses several aspects tailored to the specific requirements and features of the project.

Firstly, functional testing focuses on verifying that all system functionalities perform as expected. This includes testing user authentication mechanisms to make sure that users can successfully log in with valid credentials and are denied access with invalid ones. Additionally, each anomaly detection module, such as fire, car crash, fall, and fight detection, undergoes testing to confirm accurate identification of anomalies in both live video streams and recorded media. Furthermore, the integration with the PowerBI dashboard is thoroughly validated to ensure accurate visualization of data on road accidents and violence in public places.

Integration testing is crucial to assess the seamless integration of various system components. This involves testing the interaction between the user interface and backend services to check that user actions trigger appropriate backend processes, such as anomaly detection. Additionally, the integration with external APIs, particularly the PowerBI dashboard, is validated to ensure the accurate retrieval and display of data.

Performance testing evaluates the system's performance under different conditions. Load testing assesses the system's response to varying user loads, while response time measurements ensure prompt handling of user requests. Resource utilization is monitored to ensure optimal system performance and scalability, particularly during peak usage periods.

Security testing is paramount to safeguard sensitive data and ensure secure access to system functionalities. Data protection measures, including encryption of user credentials, are assessed for robustness. Access control mechanisms are tested to prevent unauthorized access, and vulnerability assessments are conducted to identify and address potential security threats.

Usability testing focuses on assessing the user interface for intuitiveness, navigation ease, and accessibility. Error handling mechanisms are tested to ensure clear and informative error messages if there are invalid inputs or system errors.

Reliability and scalability testing evaluate the system's ability to handle increasing loads without performance degradation and its ability to recover gracefully from failures while maintaining data integrity and availability. This comprehensive testing approach ensures that the automated surveillance system meets user requirements, delivers accurate results, and provides a seamless and secure user experience.

Chapter 7: -

RESULTS

7.1 FIRE DETECTION

Fig 7.1.1 Fire Detection Test Batch Images

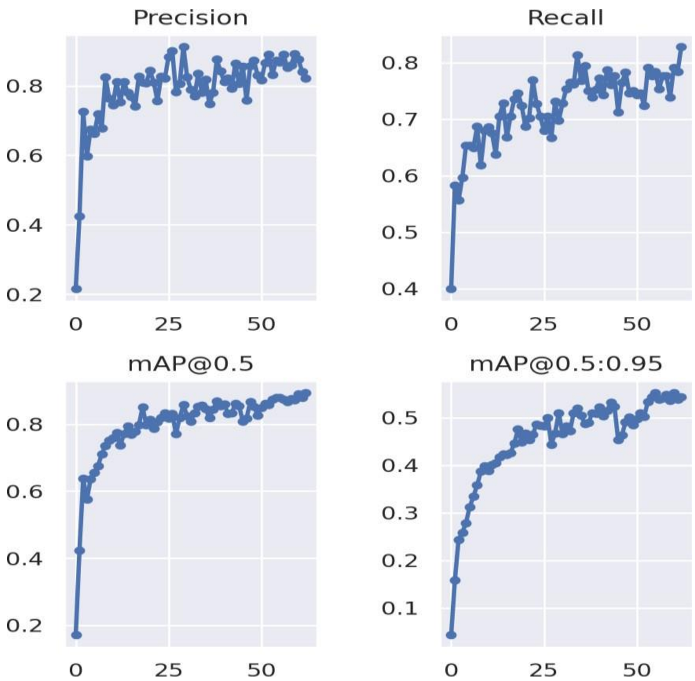


Fig 7.1.2 Fire Detection MAP Curve and PR Curves

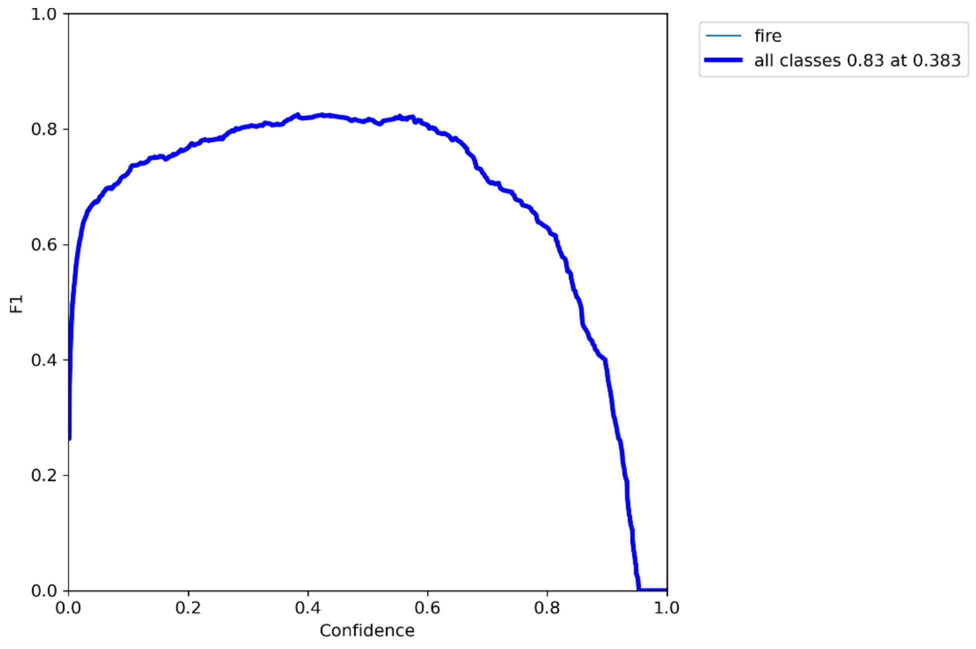


Fig 7.1.3 Fire Detection F1 Score Curve

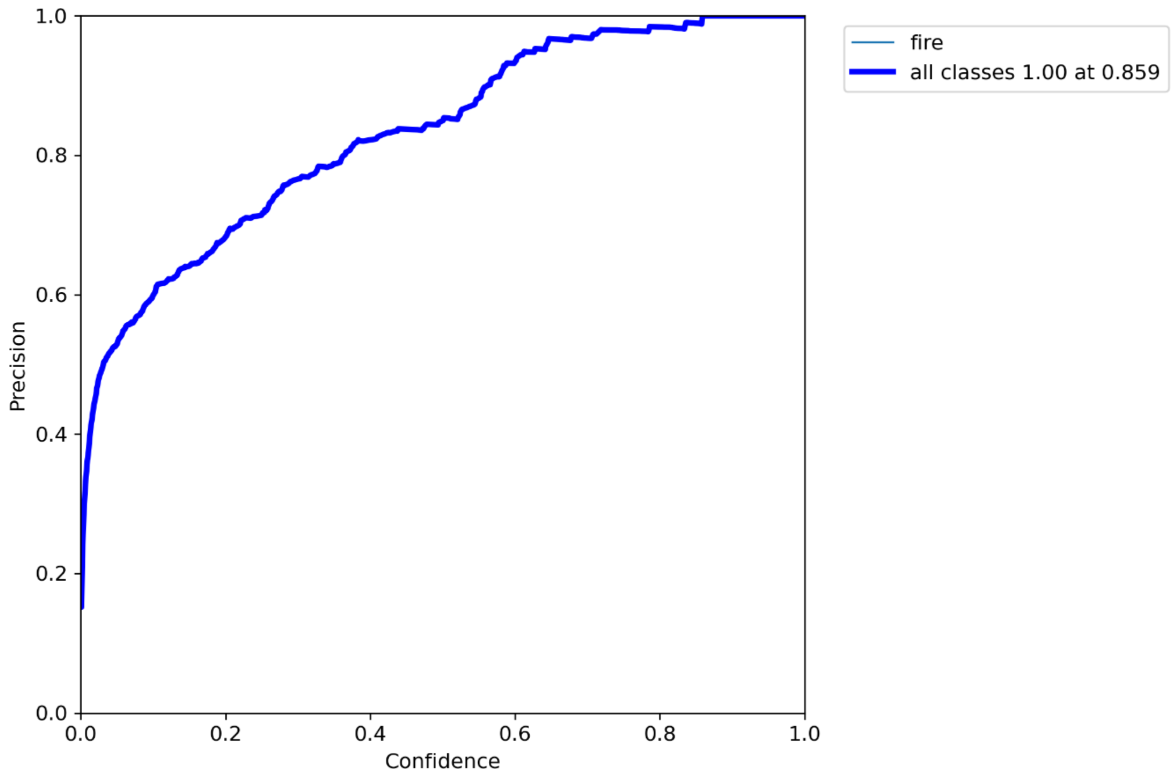


Fig 7.1.4 Fire Detection Precision Curve

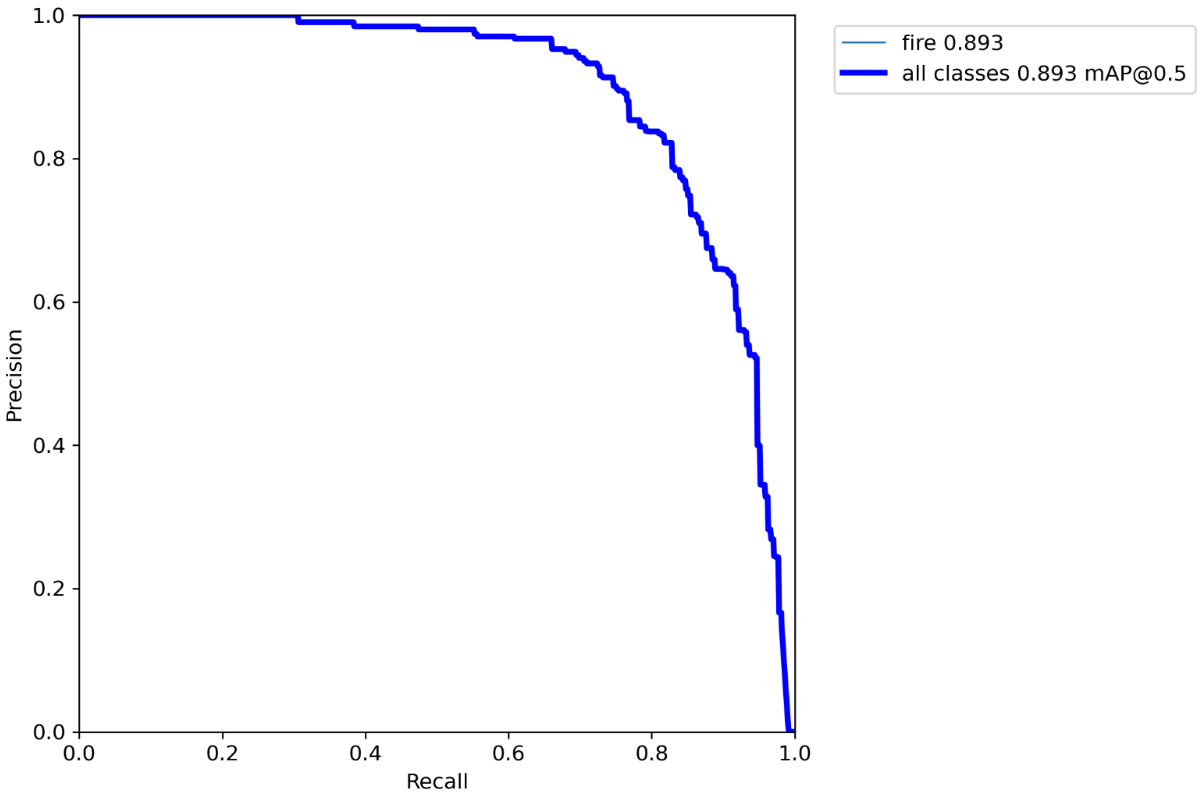


Fig 7.1.5 Fire Detection PR Curve

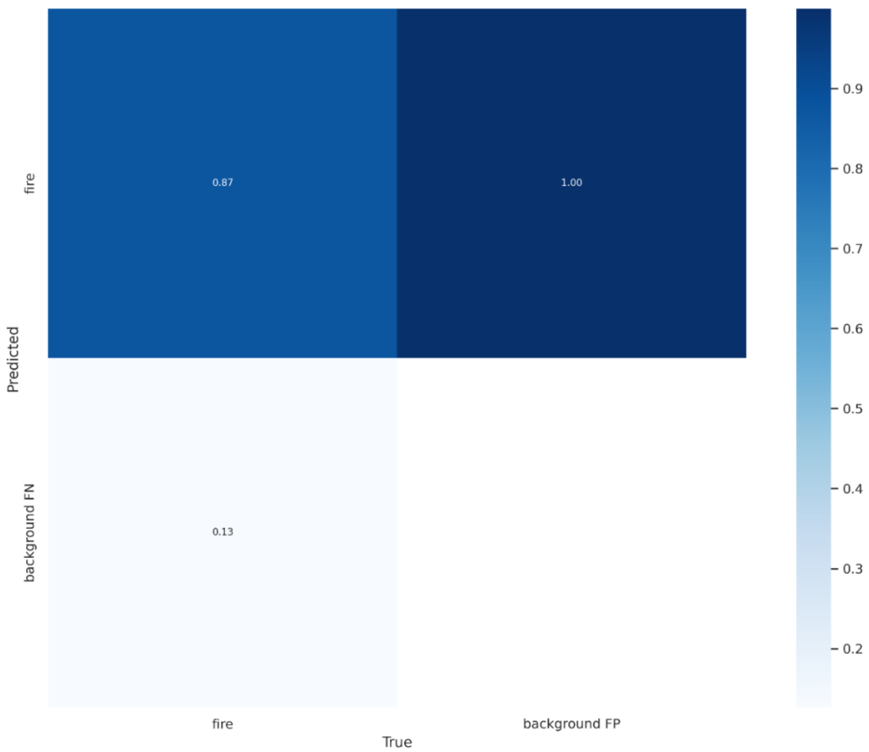


Fig 7.1.6 Fire Detection Confusion Matrix

7.2 CAR CRASH DETECTION

Fig 7.2.1 Car Crash Detection Test Batch Images

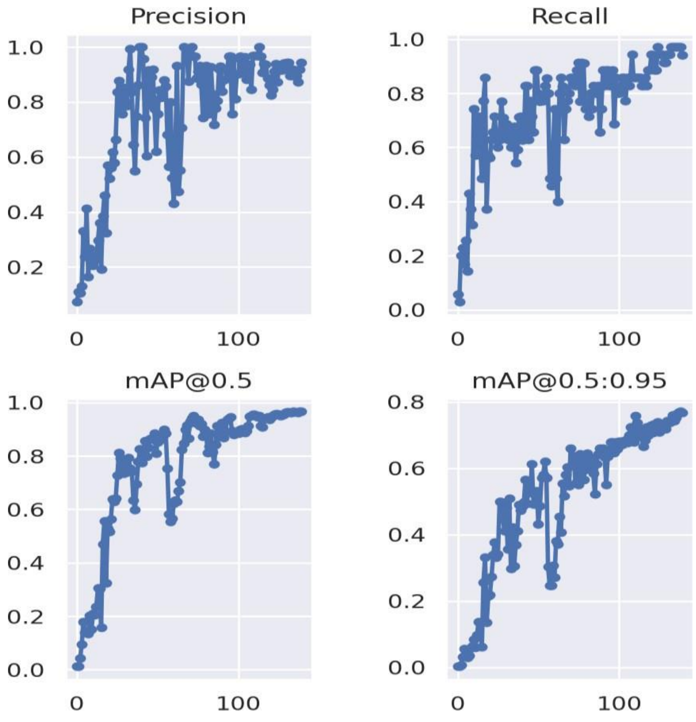


Fig 7.2.2 Car Crash Detection MAP Curve and PR Curves

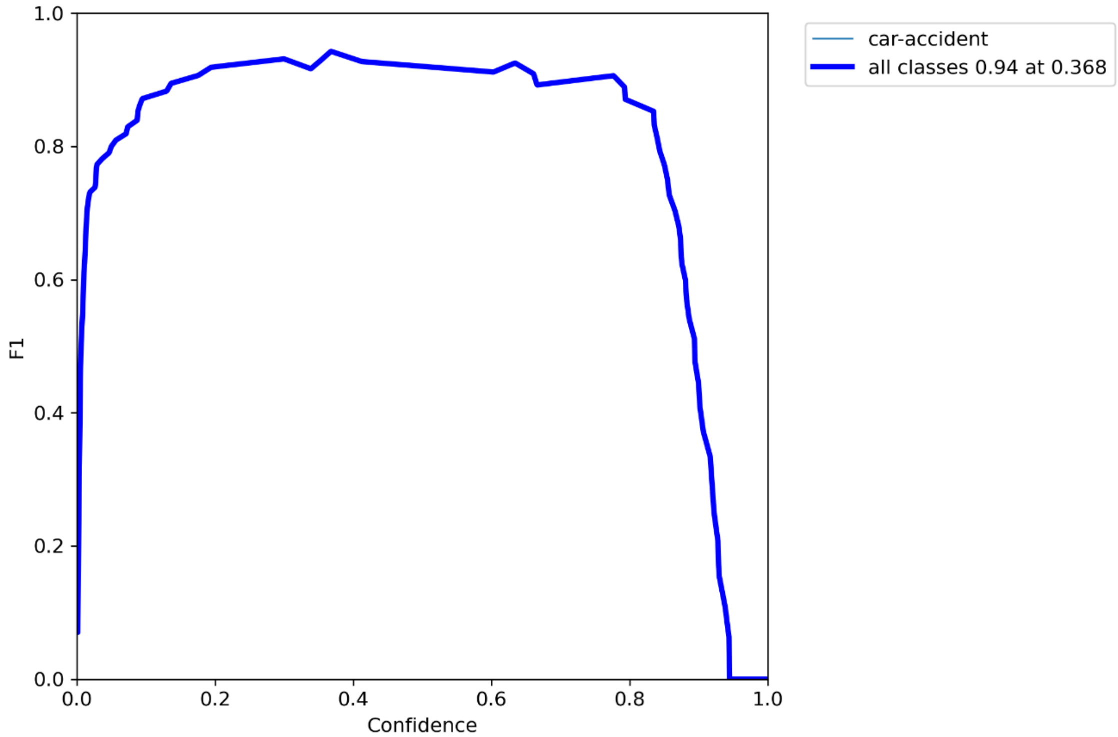


Fig 7.2.3 Car Crash Detection F1 Curve

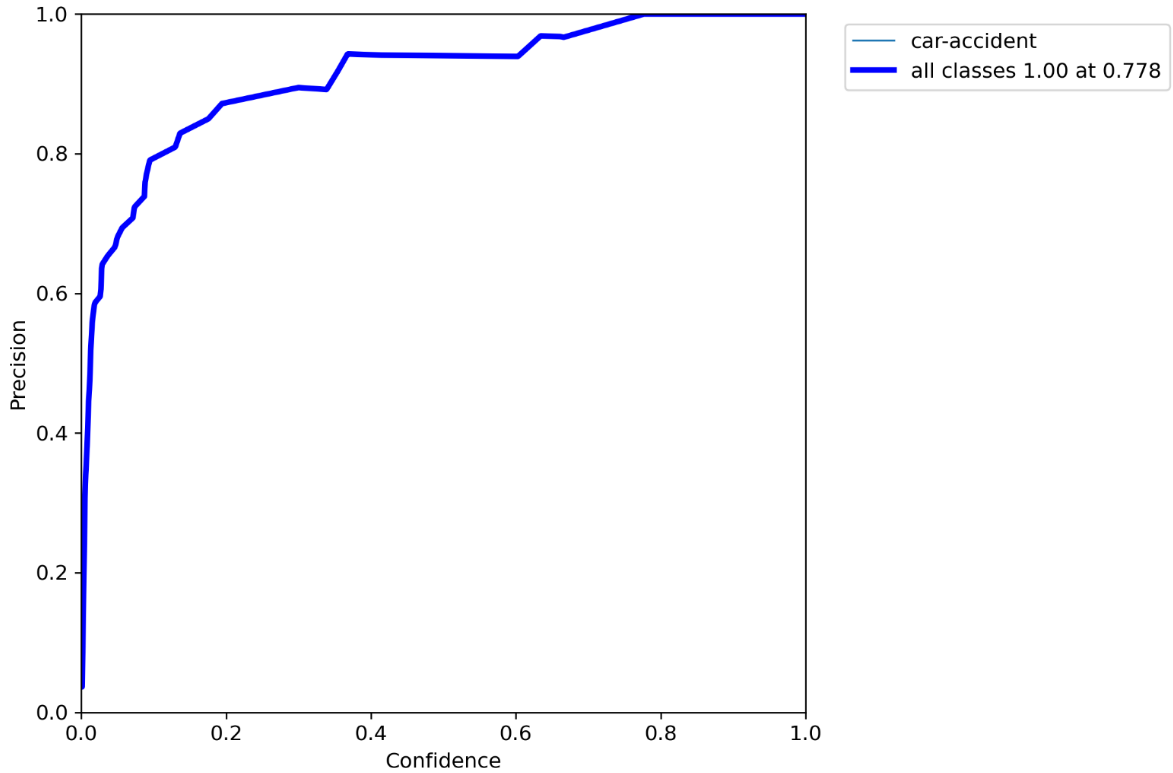


Fig 7.2.4 Car Crash Detection Precision Curve

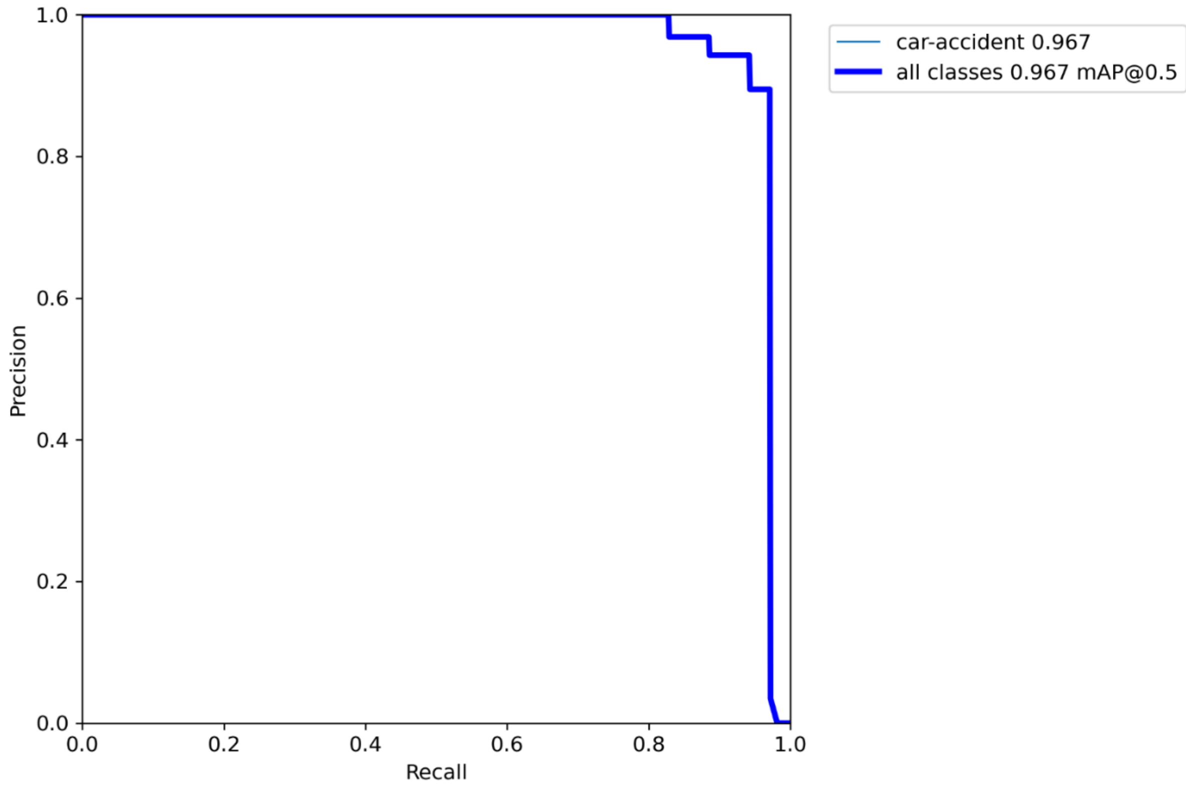


Fig 7.2.5 Car Crash Detection PR Curve

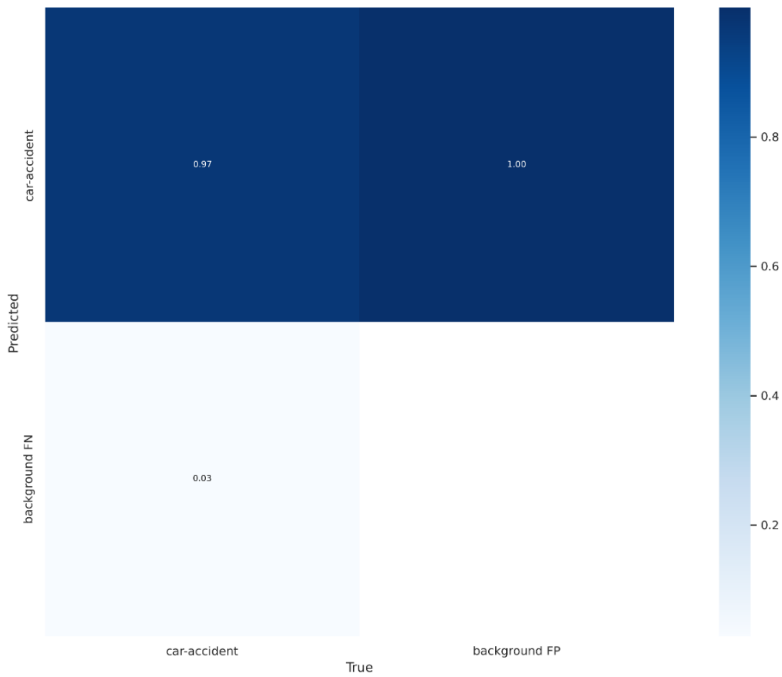


Fig 7.2.6 Car Crash Detection Confusion Matrix

7.3 FIGHT DETECTION

Fig 7.3.1 Fight Detection Test Batch Images

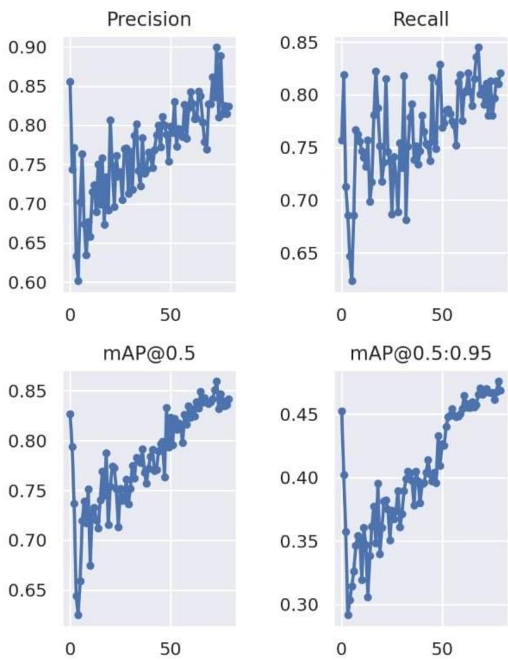


Fig 7.3.2 Fight Detection MAP Curve and PR Curves

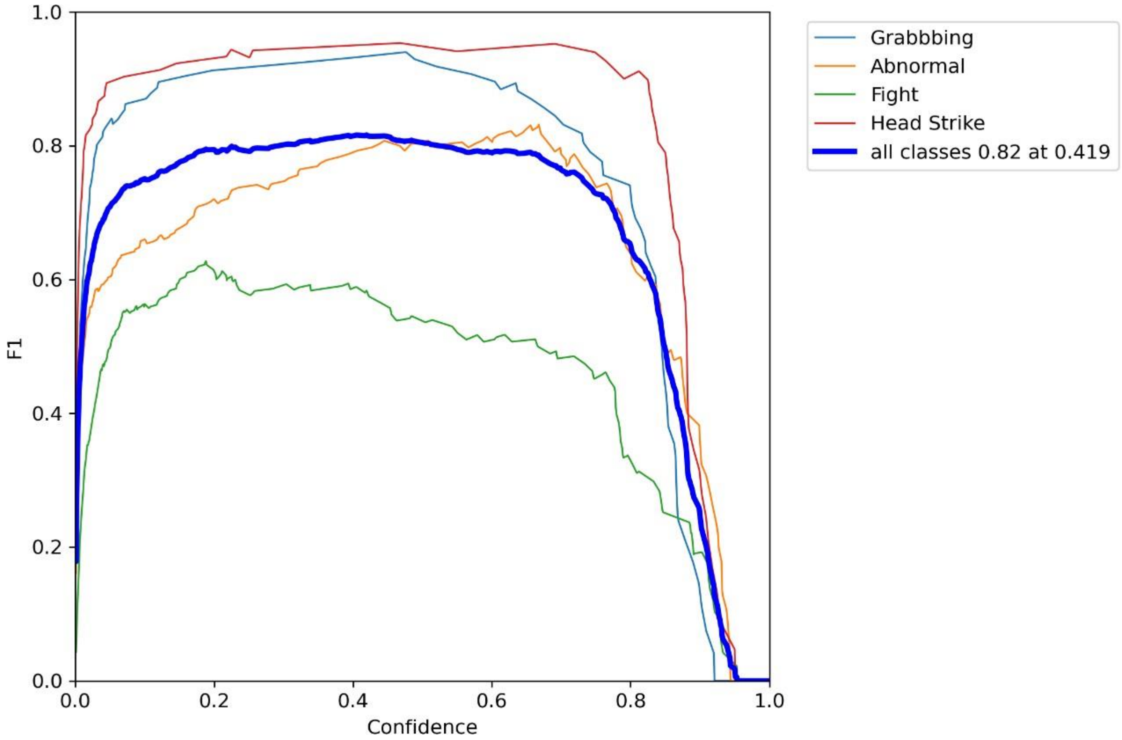


Fig 7.3.3 Fight Detection F1 Score Curve

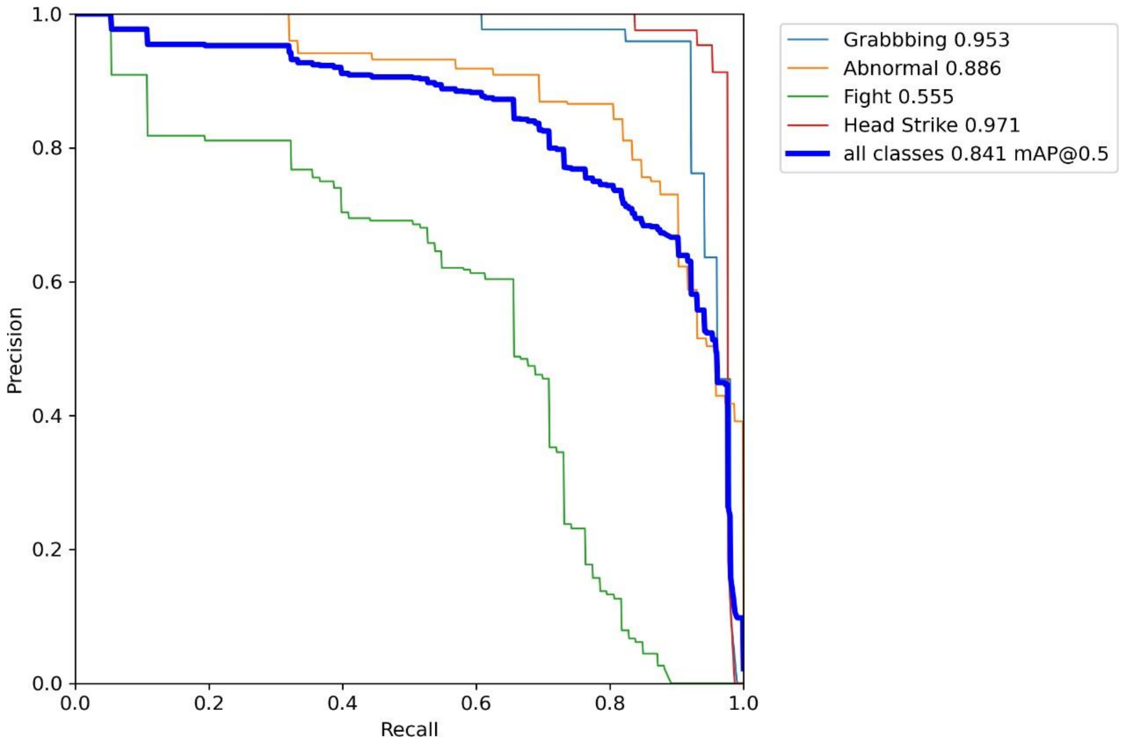


Fig 7.3.4 Fight Detection PR Curve

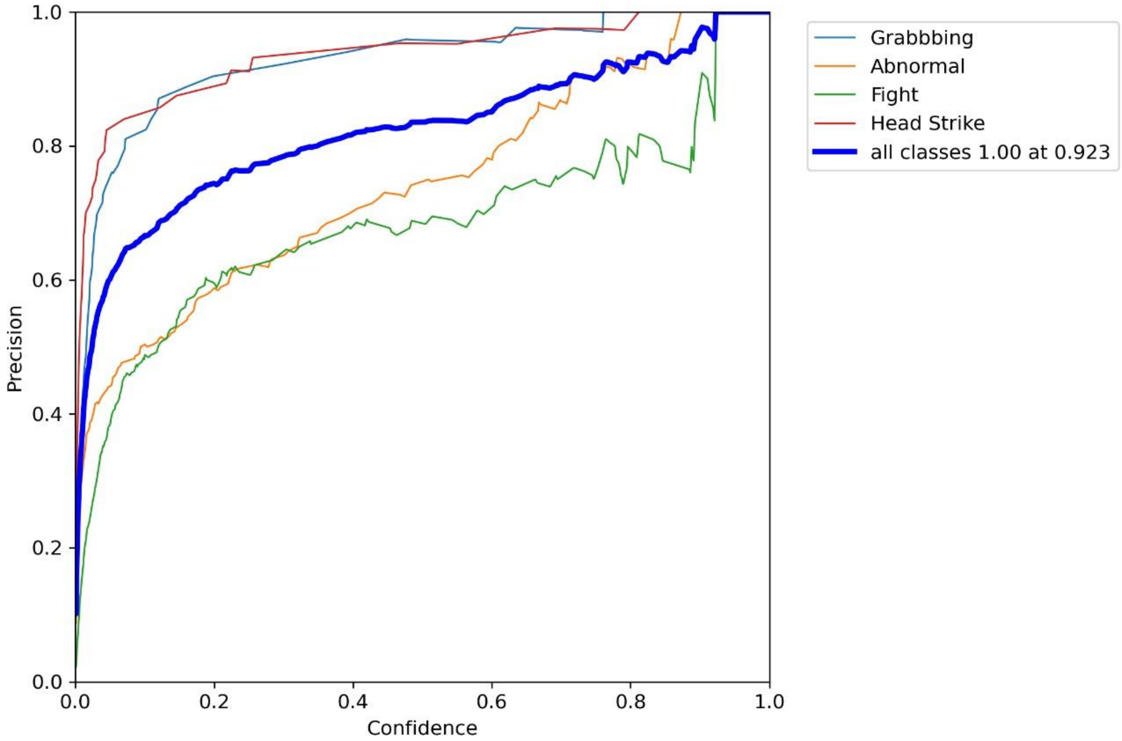


Fig 7.3.5 Fight Detection Precision Curve

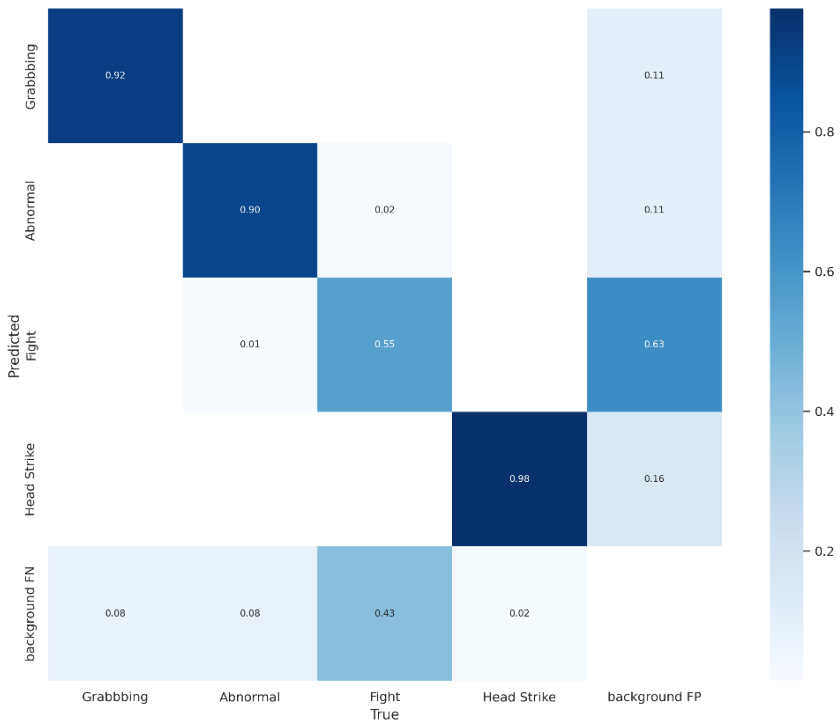


Fig 7.3.6 Fight Detection Confusion Matrix

7.4 FALL DETECTION

Fig 7.4.1 Fall Detection Test Batch Images

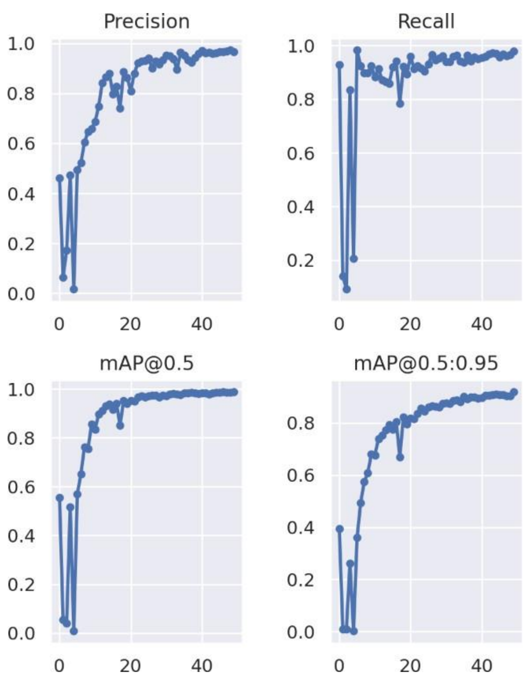


Fig 7.4.2 Fall Detection MAP Curve and PR Curve

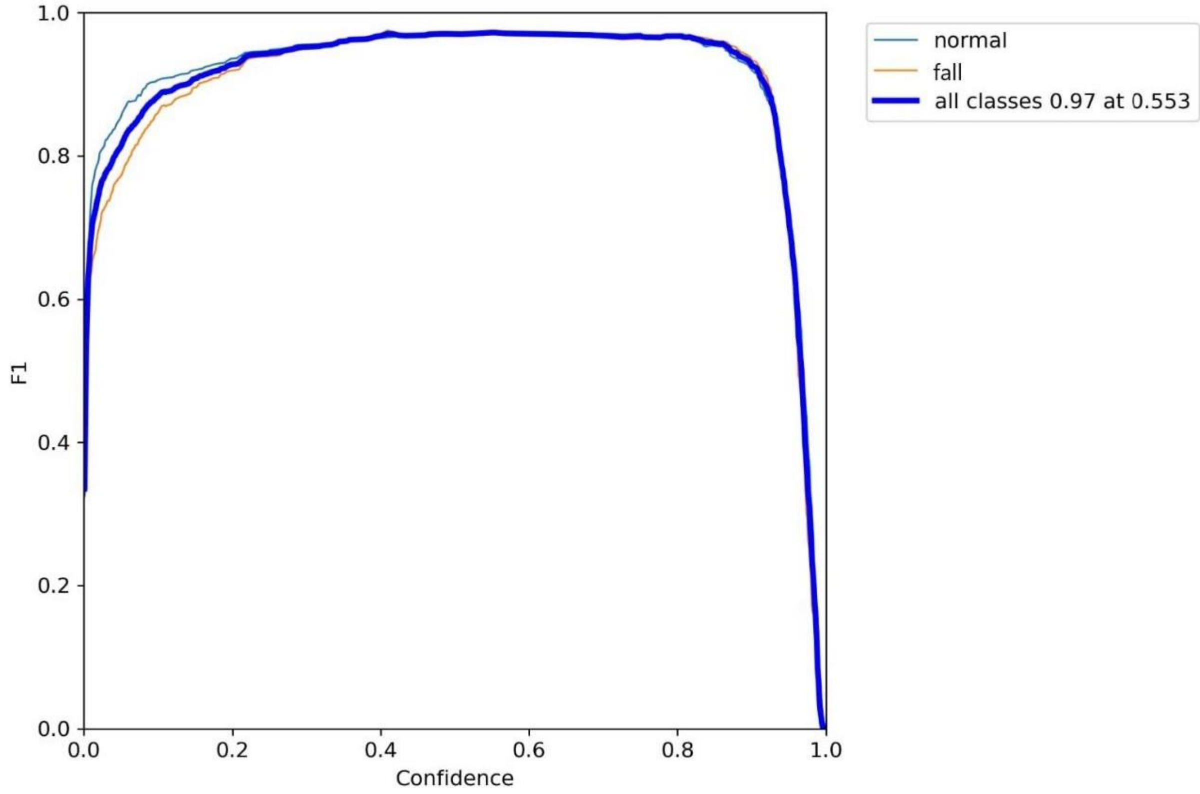


Fig 7.4.3 Fall Detection F1 Score Curve

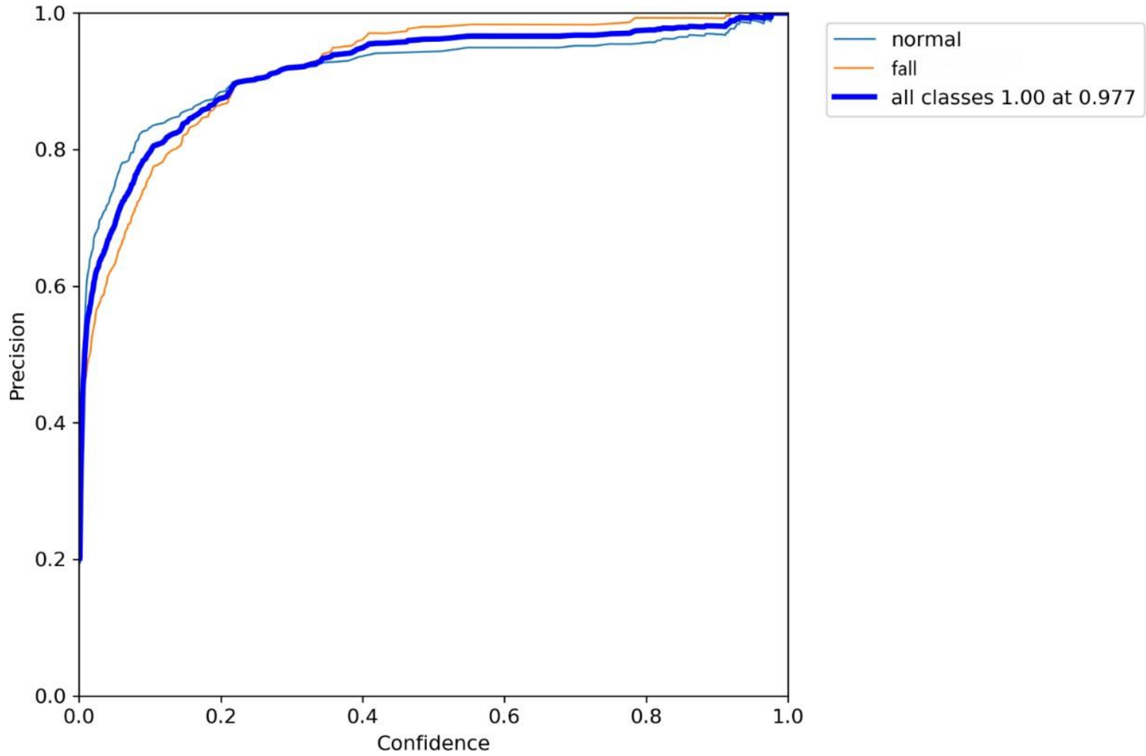


Fig 7.4.4 Fall Detection Precision Curve

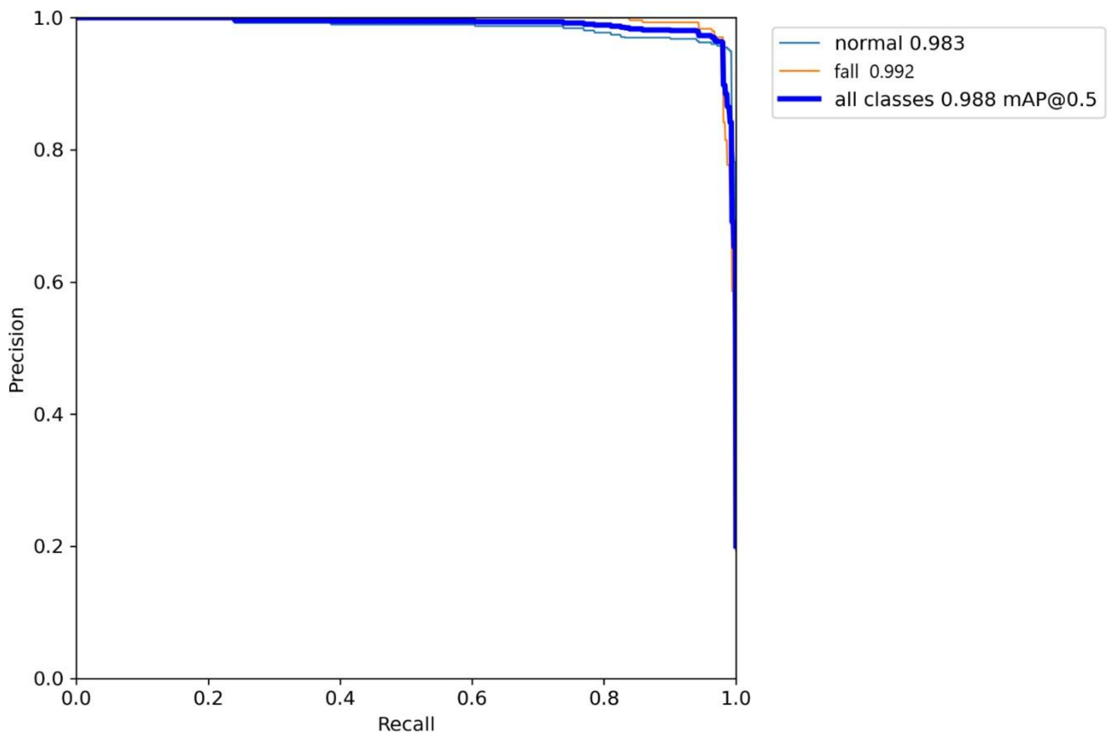


Fig 7.4.5 Fall Detection PR Curve

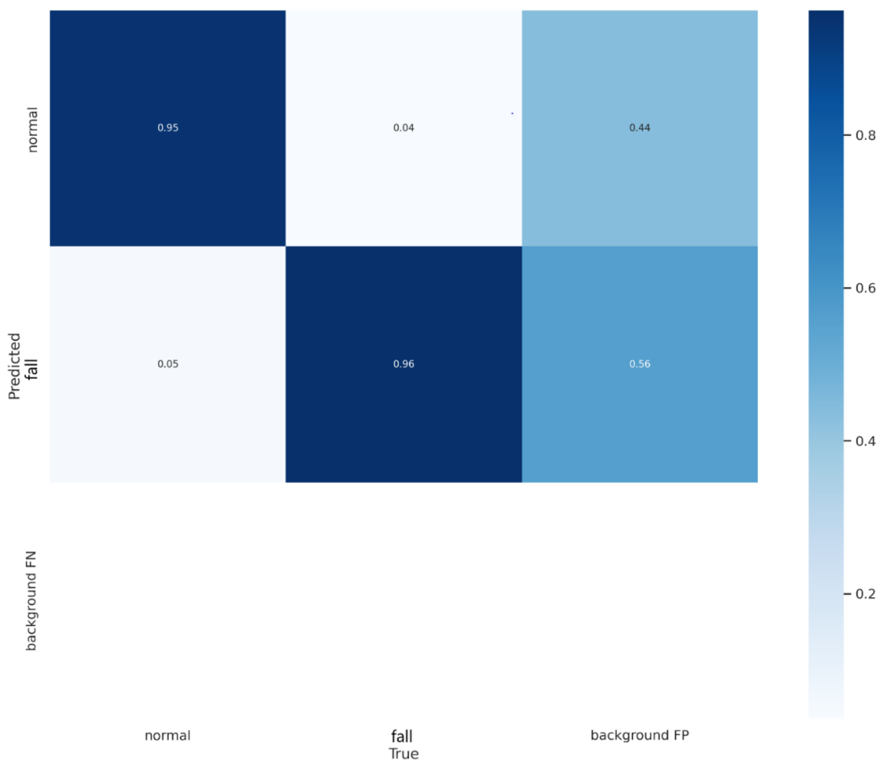


Fig 7.4.6 Fall Detection Confusion Matrix

Chapter 8: -

CONCLUSION

In conclusion, the implementation of YOLOv7 PyTorch models trained specifically for fire detection, car crash detection, fight detection, and fall detection marks a significant advancement in automated surveillance systems. Through meticulous preprocessing of data images and fine-tuning of hyperparameters, these models demonstrate remarkable accuracy and efficiency in identifying critical anomalies in real-time surveillance footage.

The utilization of transfer learning techniques further enhances the models' capabilities by leveraging pre-trained features and adapting them to the detection of specific anomalies. This way we not only accelerate the training process but also ensures robust performance across diverse surveillance environments and scenarios.

By integrating these specialized modules into the surveillance system, public safety and security are greatly enhanced. Timely detection of fire incidents, car crashes, fights, and falls enables swift response and intervention, mitigating potential risks and minimizing harm to individuals and property.

Moreover, the versatility and scalability of the YOLOv7 PyTorch models, coupled with their seamless integration with Flask for backend management, facilitate easy deployment and maintenance of the surveillance system. With the support of PostgreSQL for data management and storage, the system ensures the integrity and accessibility of critical information for analysis and optimization.

In essence, the implementation of these YOLOv7 PyTorch models represents a paradigm shift in automated surveillance, ushering in a new era of proactive incident detection and response. Through continuous refinement and innovation, these models promise to further elevate state of surveillance systems making public places more safe and secure.

REFRENCES

[1] Huu-Thanh Duong,Viet-Tuan Le and Vinh Truong Hoang. “Deep Learning-Based Anomaly Detection in Video Surveillance: A Survey,MDPI ,May 2023”

[2] Ruben J Franklin, Mohana, Vidyashree Dabbagol. “Anomaly Detection in Videos for Video Surveillance Applications using Neural Networks,ResearchGate, January 2020”

[3] Harshitha G, Hemanth Kumar A. “Anomaly Detection in Video Surveillance using Deep Learning, IJRCRT, August 2022”

[4] M. Elarbi-Boudihir, Khalid A. Al-Shalfan. “INTELLIGENT VIDEO SURVEILLANCE SYSTEM ARCHITECTURE FOR ABNORMAL ACTIVITY DETECTION, Acadameia Aug 2021”

[5] Devashree R. Patrikar Mayur Rajaram Parate.”Anomaly detection using edge computing in video surveillance system: review,Springer,March 2022”

[6] Chongke Wu, Sicong Shao, Cihan Tunc, Salim Hariri.”Video Anomaly Detection Using Pre-Trained Deep Convolutional Neural Nets and Context Mining,ResearchGate,October 2022”

[7] Jia-Yao Su, Che-Ming Wu, Shuqun Yang. “Object-Tracking Algorithm Combining Motion Direction and Time Series ,MDPI Apr 2023”

[8] Felix M Philip, Jayakrishnan, F. Ajesh, Haseena P.”Video Anomaly Detection Using the Optimization-Enabled Deep Convolutional Neural Network,ResearchGate,April 2021”

[9] Muhammad Hussain. “YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection, MDPI ,Jun 2023”.

[10] Xuanzhao Wang, Zhengping Che, Bo Jiang, Ning Xiao, Ke Yang, Jian Tang, Jieping Ye, Jingyu Wang, Qi Qi.”Robust Unsupervised Video Anomaly Detection by Multi-Path Frame Prediction,Axqi,May-2020

[11] Sabrina Aberkane, Mohamed Elarbi-Boudihir. “Deep Reinforcement Learning-based anomaly detection for Video Surveillance,Informatica,April 2022”.

[12] Abdelhafid Berroukham, Khalid Housni, Mohammed Lahraichi, Idir Boulfrifi.”Deep learning-based methods for anomaly detection in video surveillance: a review,ResearchGate,January 2023

[13] Aswathy K. Cherian, E. Poovammal . “Anomaly Detection in Real-Time Surveillance Videos Using Deep Learning,X-MOL,Apr 2021”.

[14] Zheyi Fan, Jianyuan Yin, Yu Song, Zhiwen Liu. “Real-time and accurate abnormal behavior detection in videos, ResearchGate, Sep 2020

[15] Abid Mehmood,”Efficient Anomaly Detection in Crowd Videos Using Pre-Trained 2D Convolutional Neural Networks,ResearchGate,October 2021.

[16] Mingzhu Huang, Jie Tao, Chang Ying Wang, Baojun Waukesha Li, Han Shen, Hancheng Yu. “Object tracking algorithm of Siamese network based on feature fusion and attention mechanism, ADS HARVARD , May 2023”.

[17] ZHIYUAN LI , JING CHEN, AND JIERAN BI. “Multiple object tracking with appearance feature prediction and similarity fusion, IEEE Access, Jan 2023”.

[18] Ms. Archana Karne, Mr. RadhaKrishna Karne, M. V. Kumar, Dr. A. Arunkumar. “Convolutional Neural Networks for Object Detection and Recognition, Research Gate , Feb 2023”.

[19] Ziyuan Lu, Zhenkun Wang. “Researches advanced in object tracking based on deep learning, ADS HARVARD, Nov 2022”.

[20] Tejonidhi R. Deshpande, Sagar U. Sapkal. “Development of Object Tracking System Utilizing Camera Movement and Deep Neural Network,IEEE XPLORE, Jul 2022”.

APPENDIX

SNAPSHOTS

Fig A.1 Home Screen Without Login

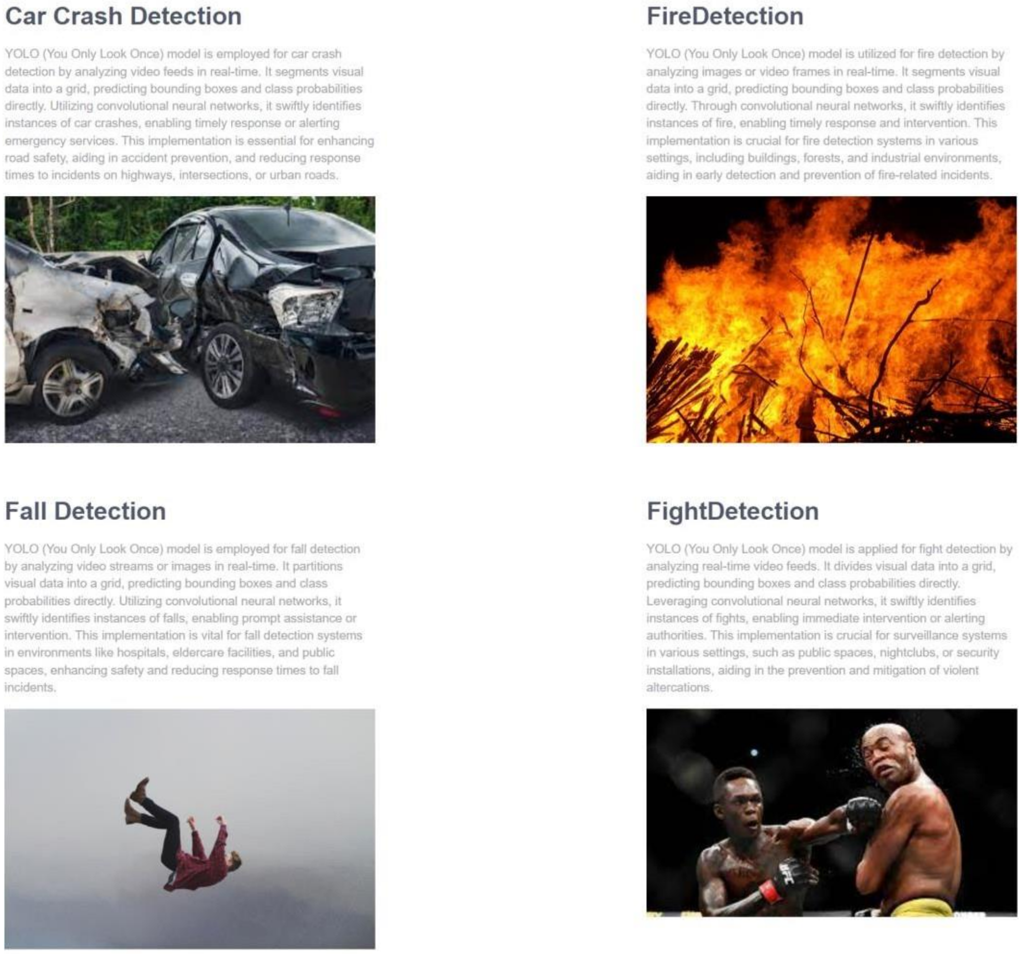


Fig A.2 Services Provided

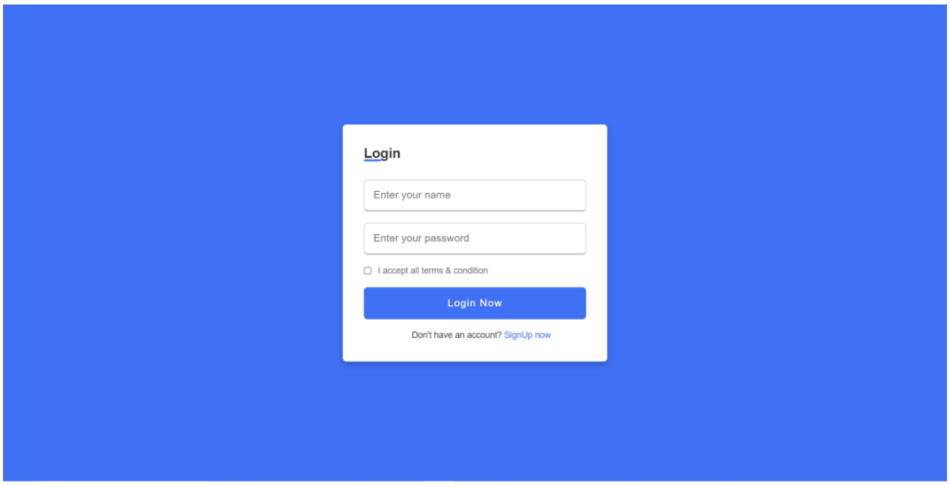


Fig A.3 Login Page

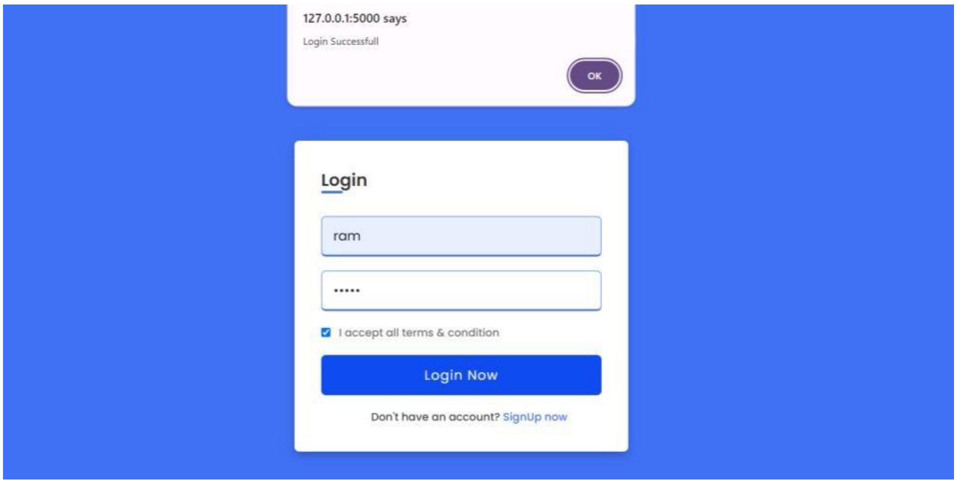


Fig A.4 Login Successful

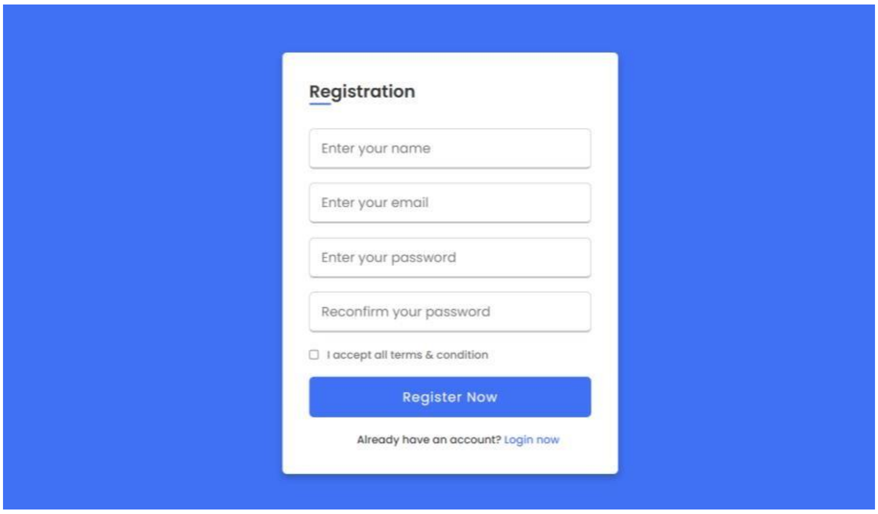


Fig A.5 Signup Page

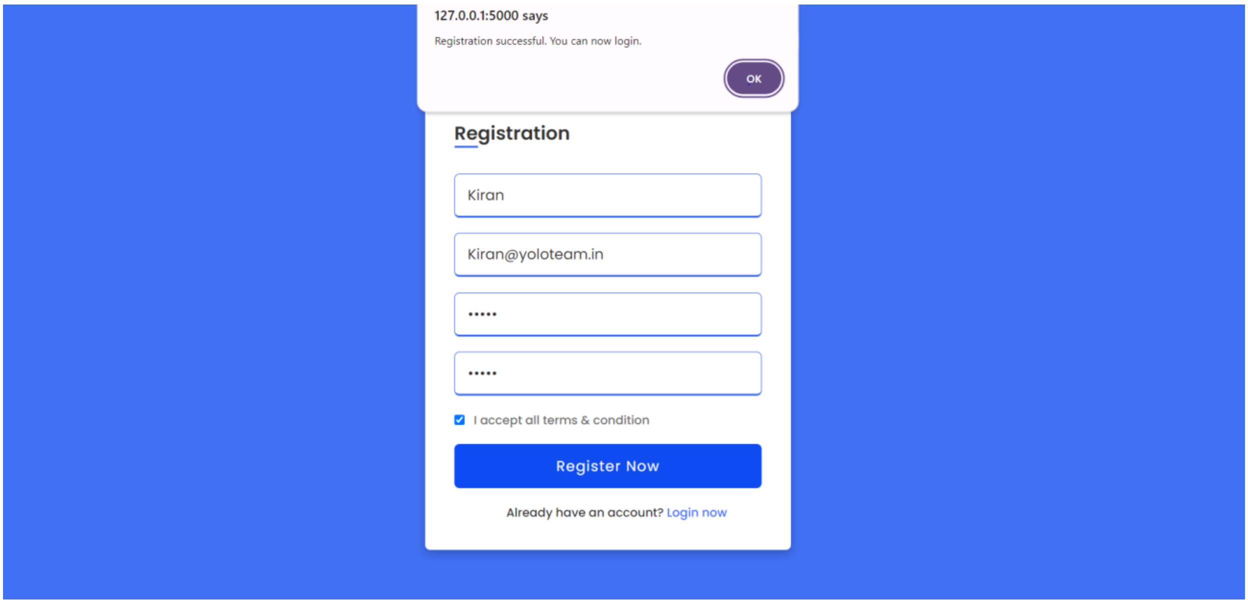


Fig A.6 New User Created Successful

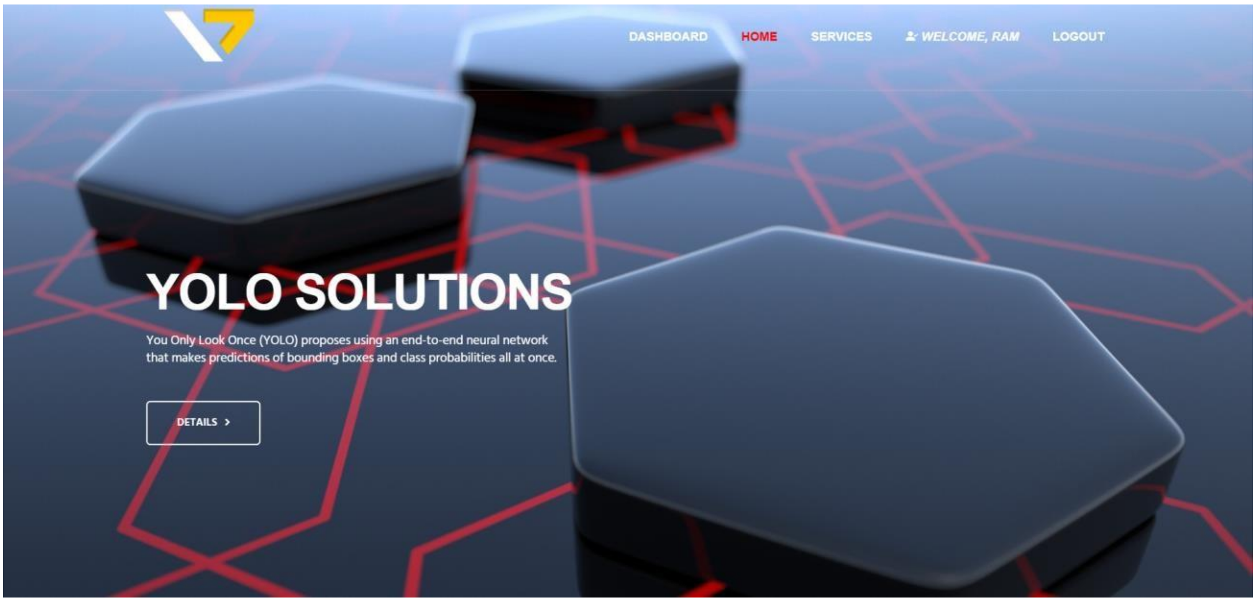


Fig A.7 Home Screen After Login

Fig A.8 Car Crash Detection Page

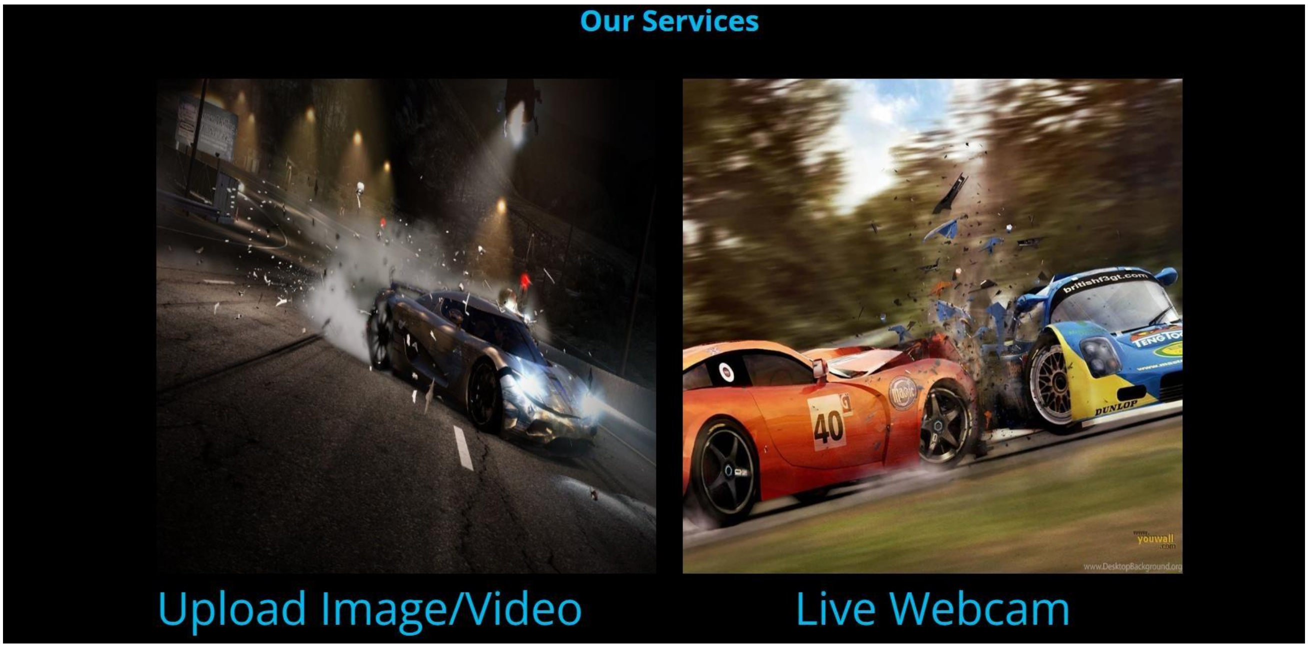


Fig A.9 Option to choose between Live or Recorded Video for Car Crash Detection

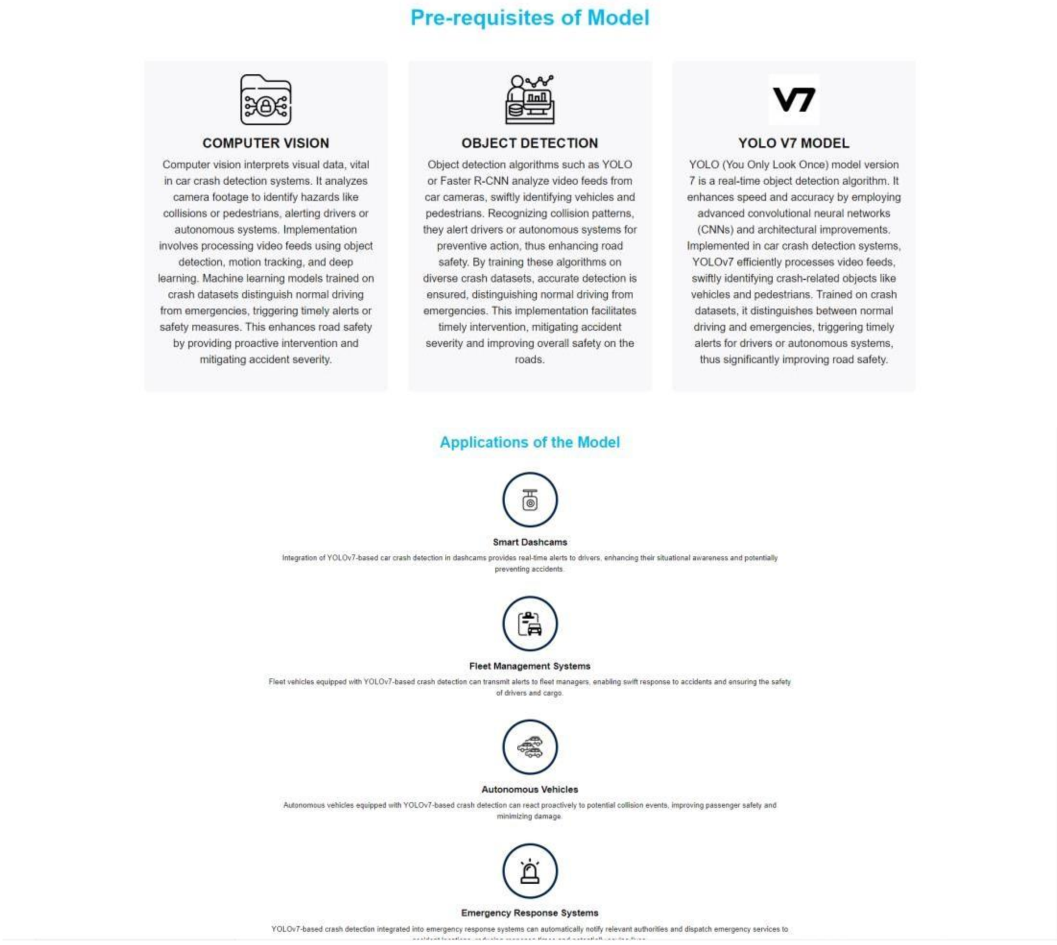


Fig A.10 Car Crash Detection Prerequisites and Applications

Fig A.11 Fire Detection Page

Fig A.12 Option to choose between Live or Recorded Video for Fire Detection

Fig A.13 Fire Detection Prerequisites and Applications

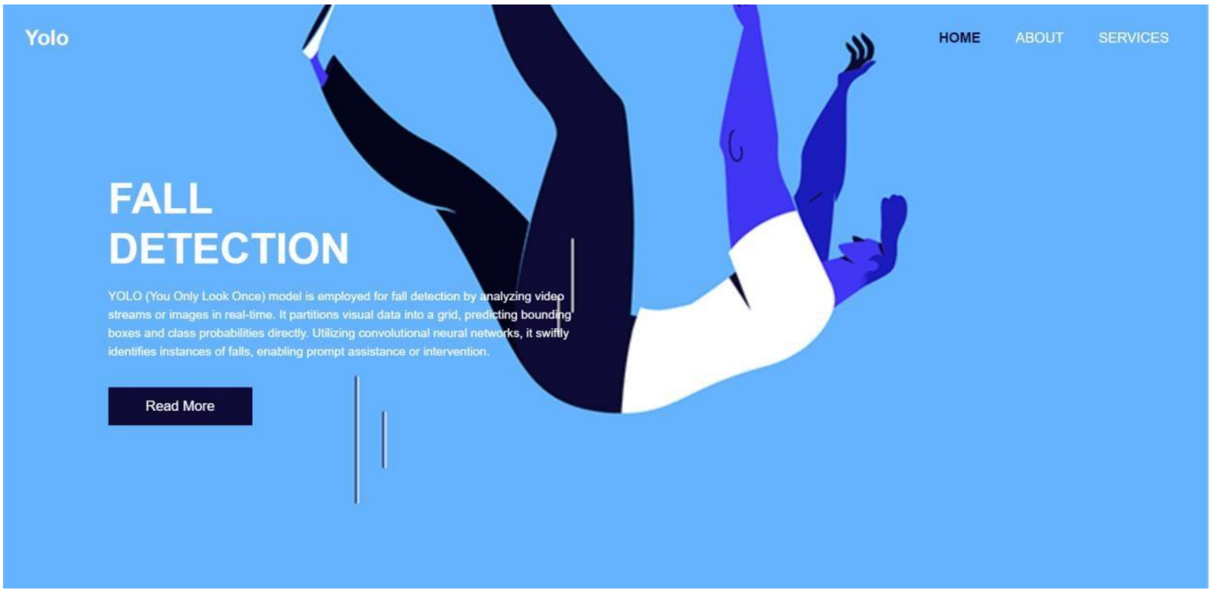


Fig A.14 Fall Detection Page

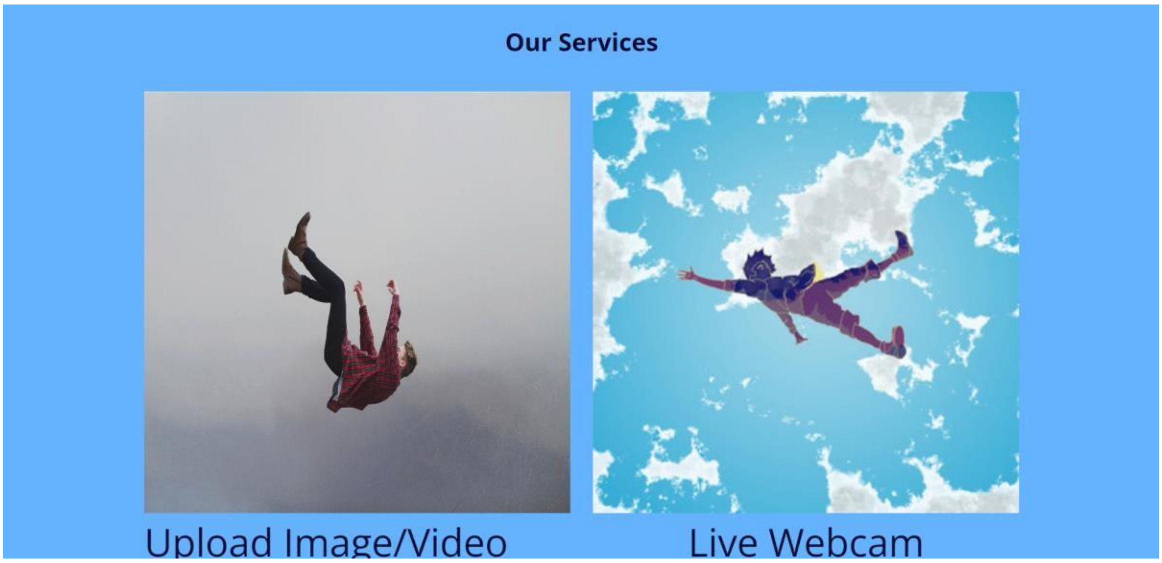


Fig A.15 Option to choose between Live or Recorded Video for Fall Detection

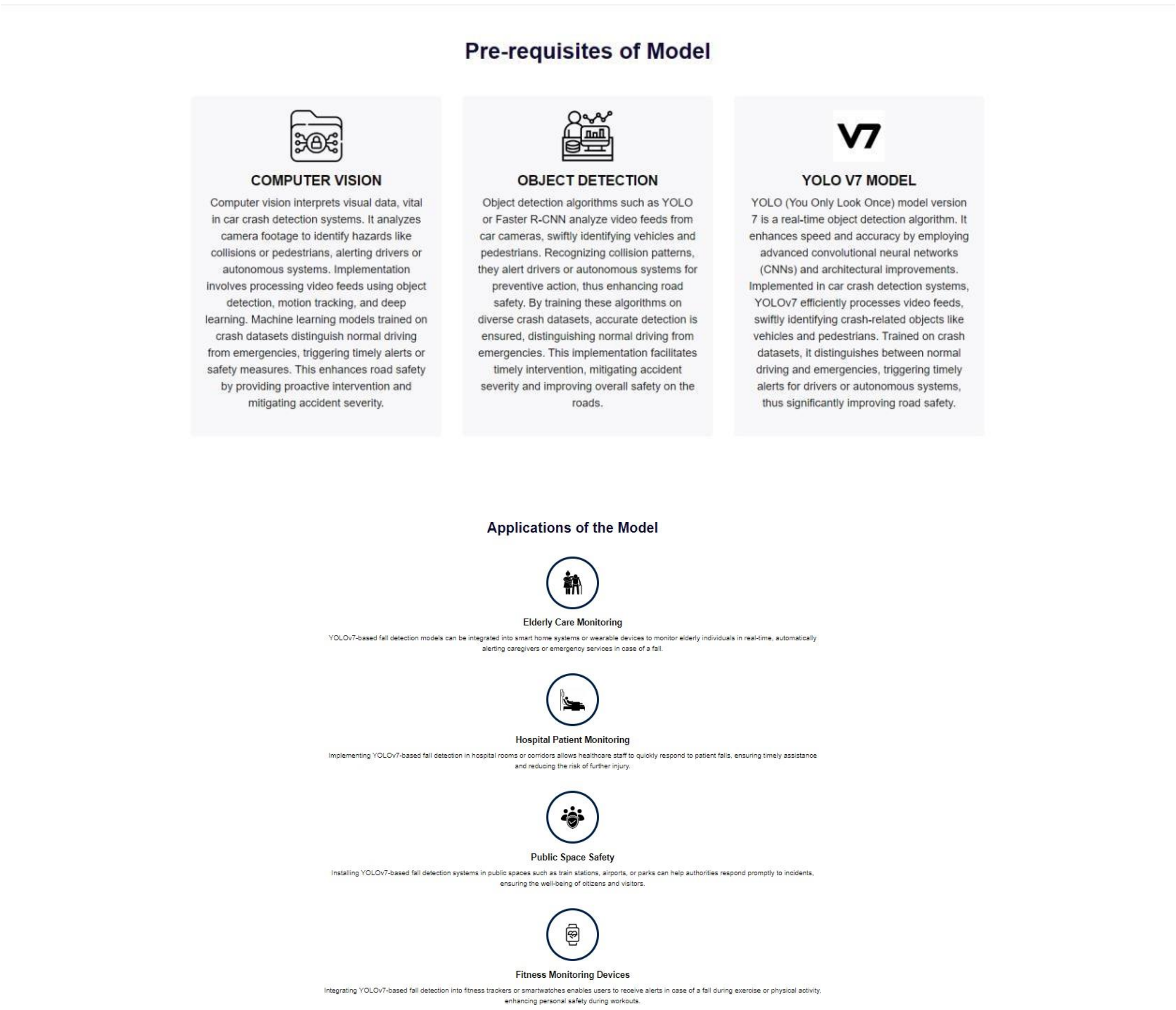


Fig A.16 Fall Detection Prerequisites and Applications

Fig A.17 Fight Detection Page

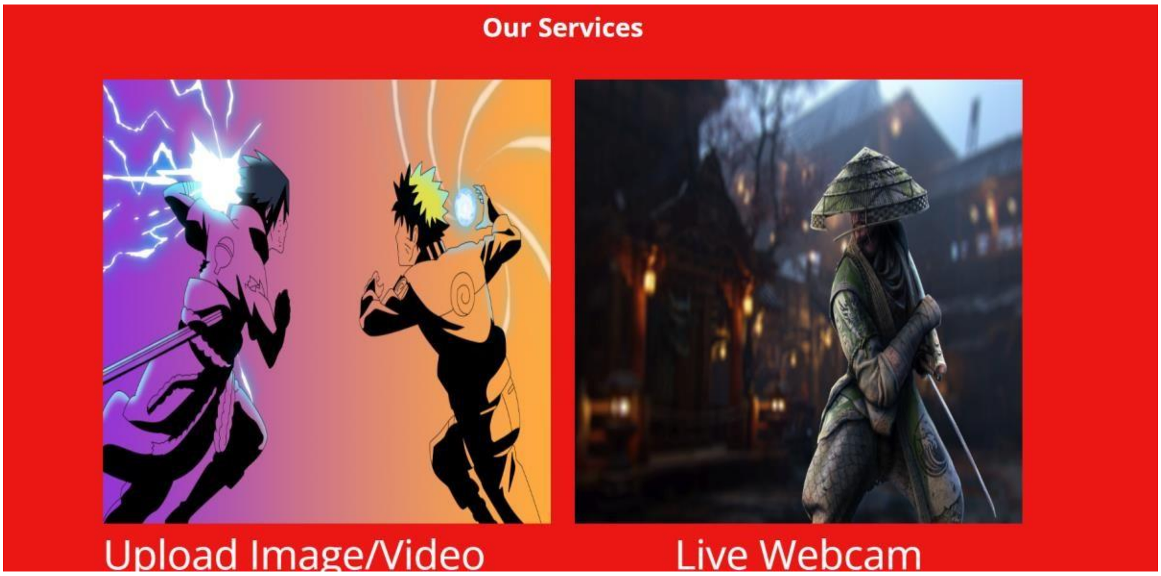


Fig A.18 Option to choose between Live or Recorded Video for Fight Detection

Fig A.19 Fight Detection Prerequisites and Applications

Fig A.20 Car Crash Anomaly 1 Detected

Fig A.21 Car Crash Anomaly 2 Detected

Fig A.22 Fire Anomaly 1 Detected

Fig A.23 Fire Anomaly 2 Detected

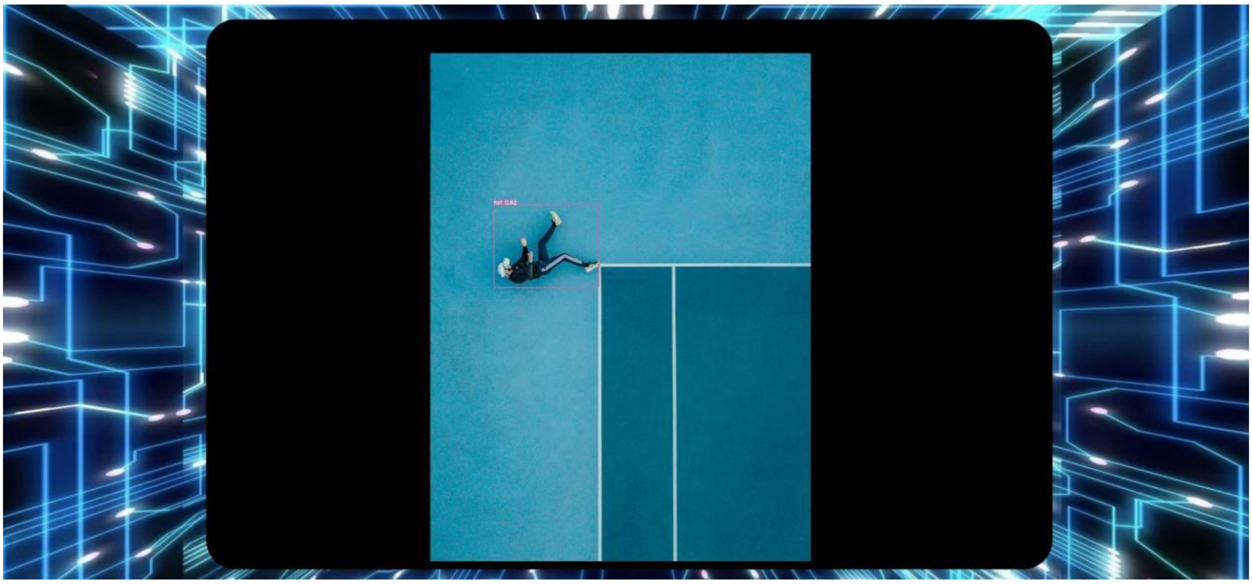


Fig A.24 Fall Anomaly 1 Detected

Fig A.25 Fall Anomaly 2 Detected

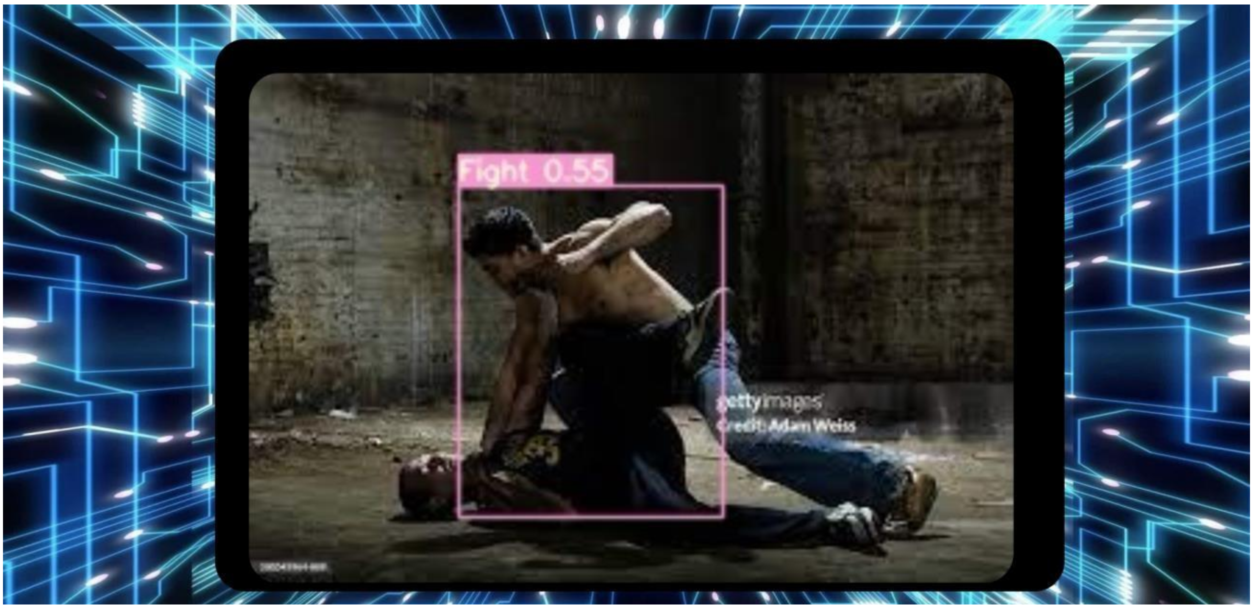


Fig A.26 Fight Anomaly 1 Detected

Fig A.27 Fight Anomaly 2 Detected

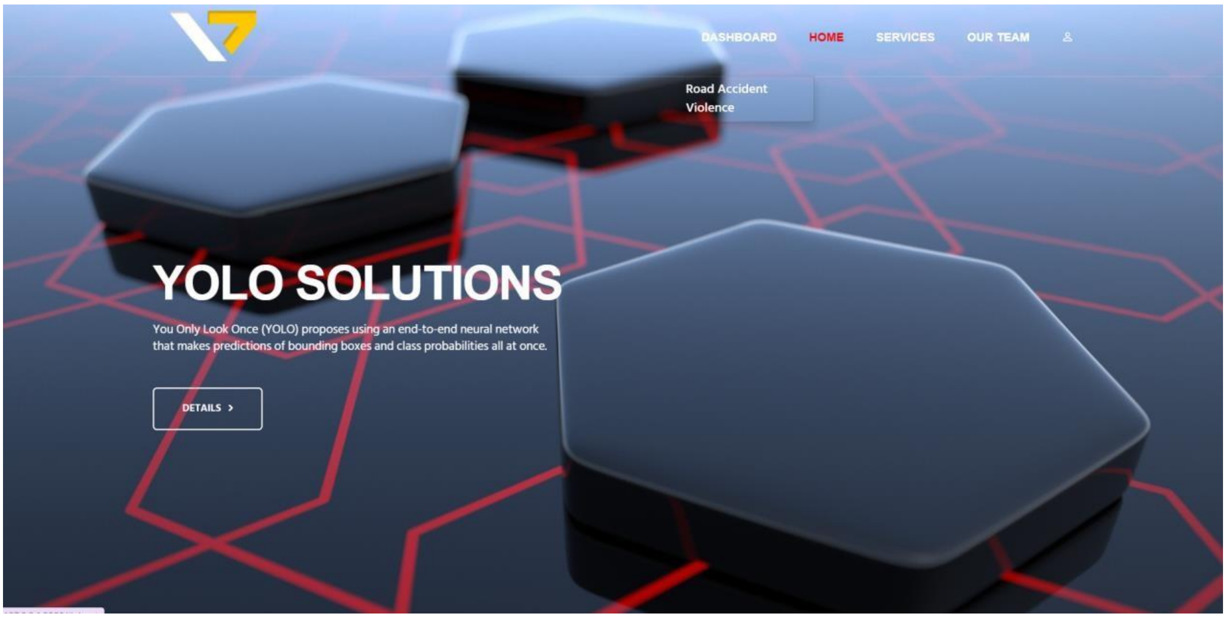


Fig A.28 Dropdown for Dashboard Visualizations

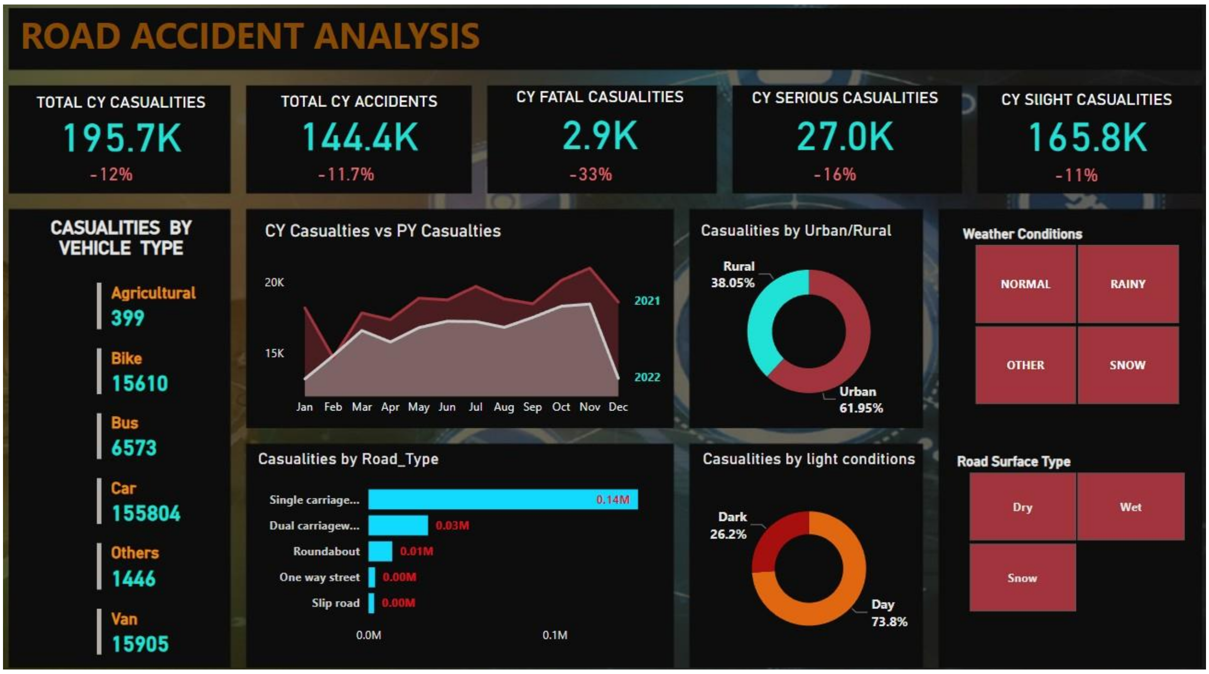


Fig A.29 Road Accident Dashboard

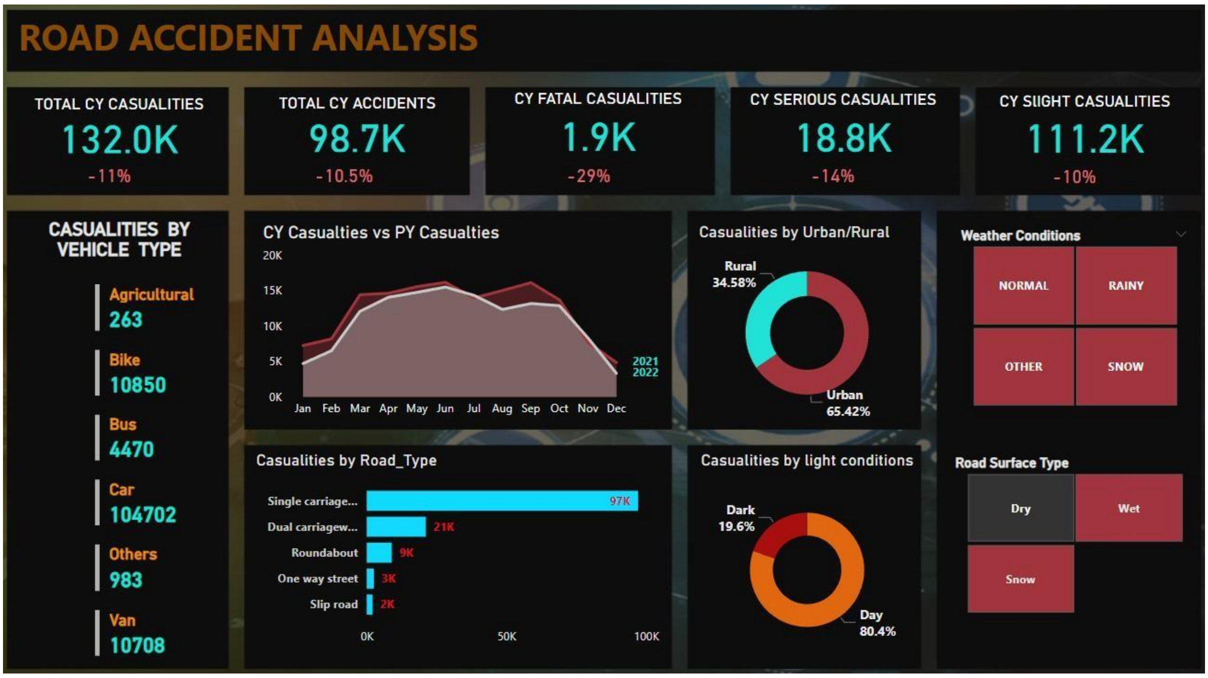


Fig A.30 Road Accident Dry Road Surface Type

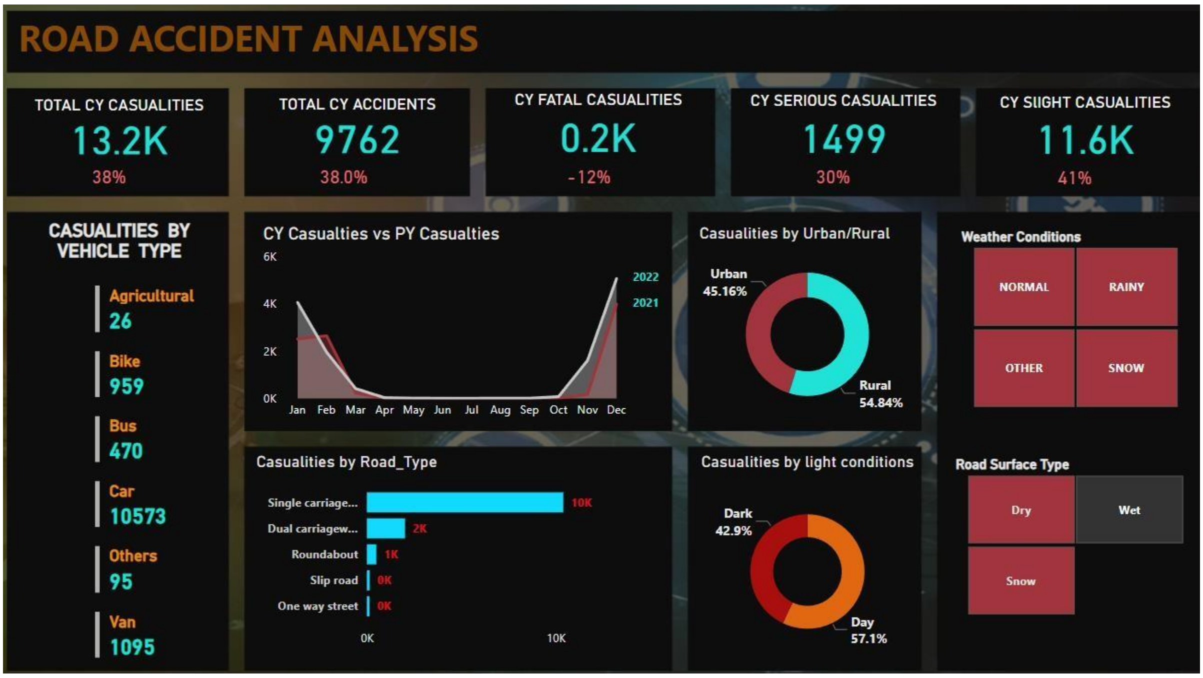


Fig A.31 Road Accident Wet Road Surface Type

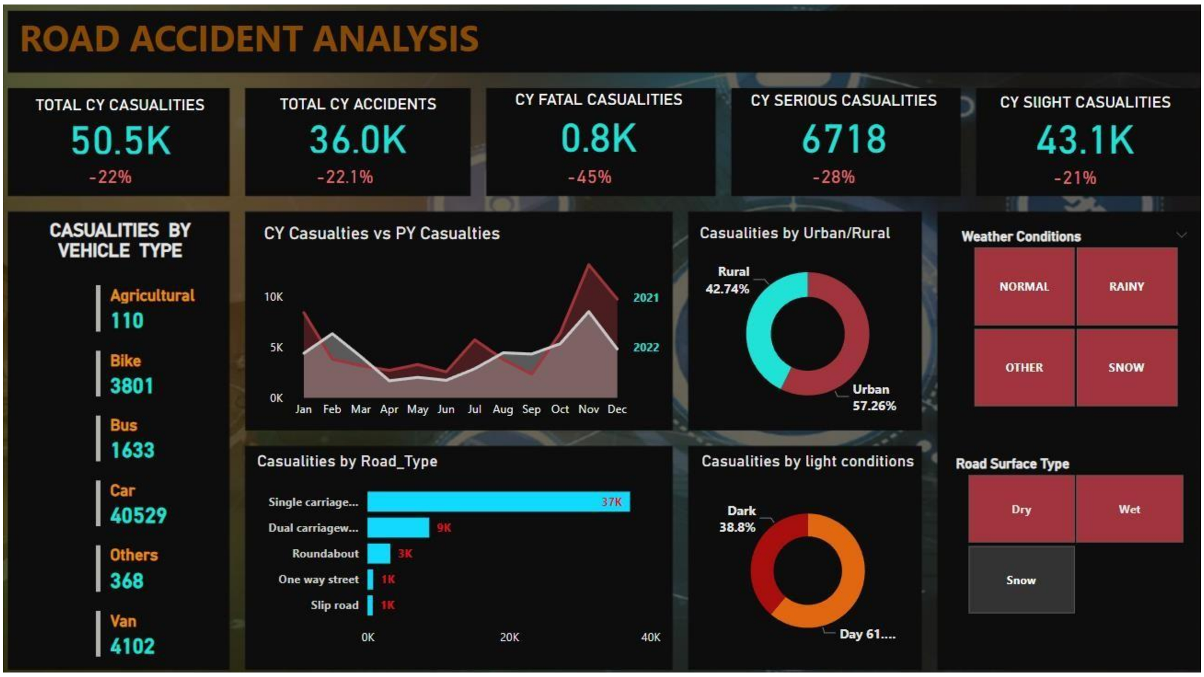


Fig A.32 Road Accident Snow Road Surface Type

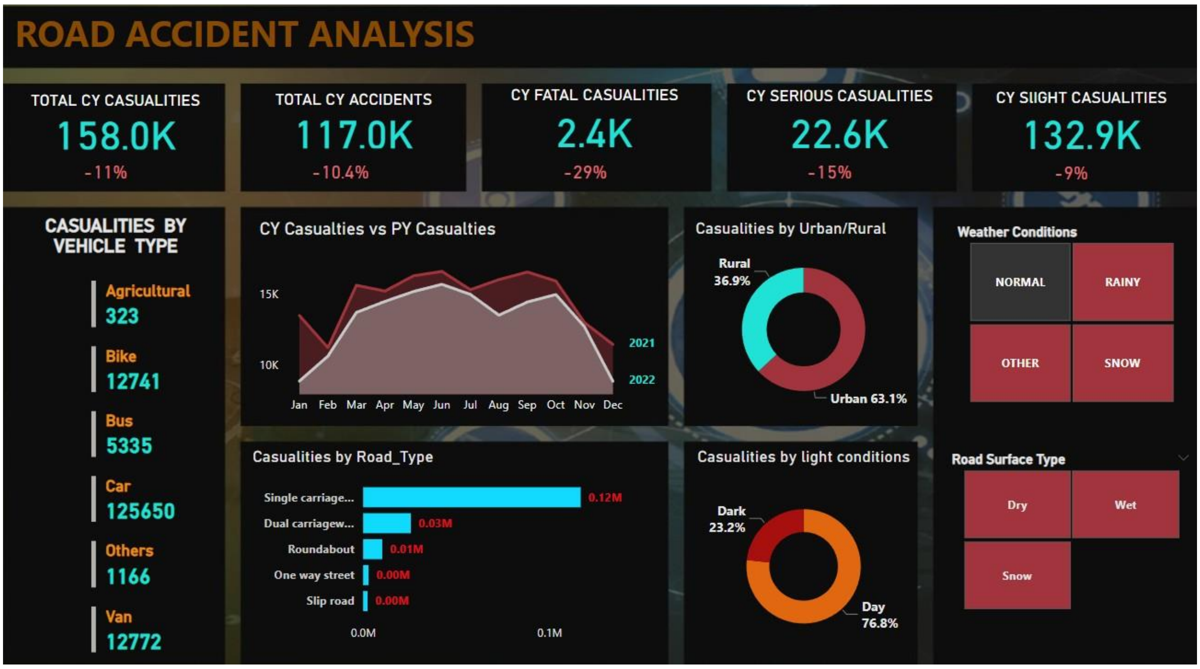


Fig A.33 Road Accident Normal Weather Condition

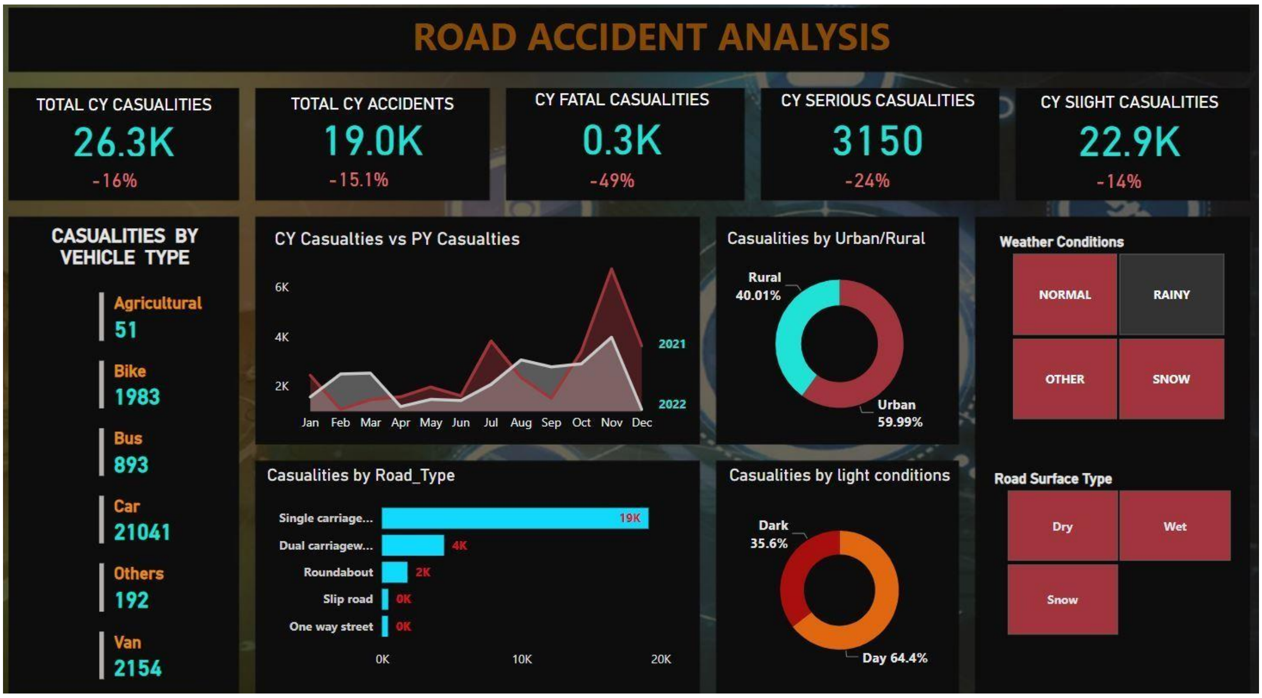


Fig A.34 Road Accident Rainy Weather Condition

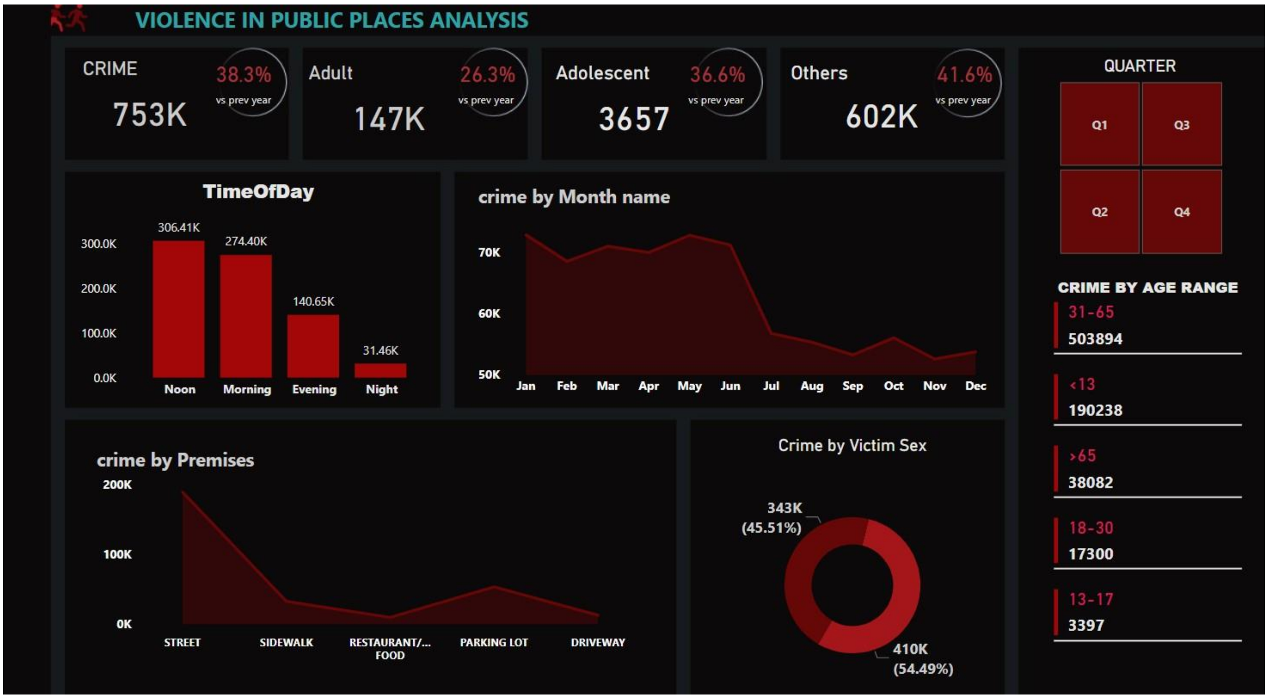


Fig A.35 Violence in Public Place Dashboard

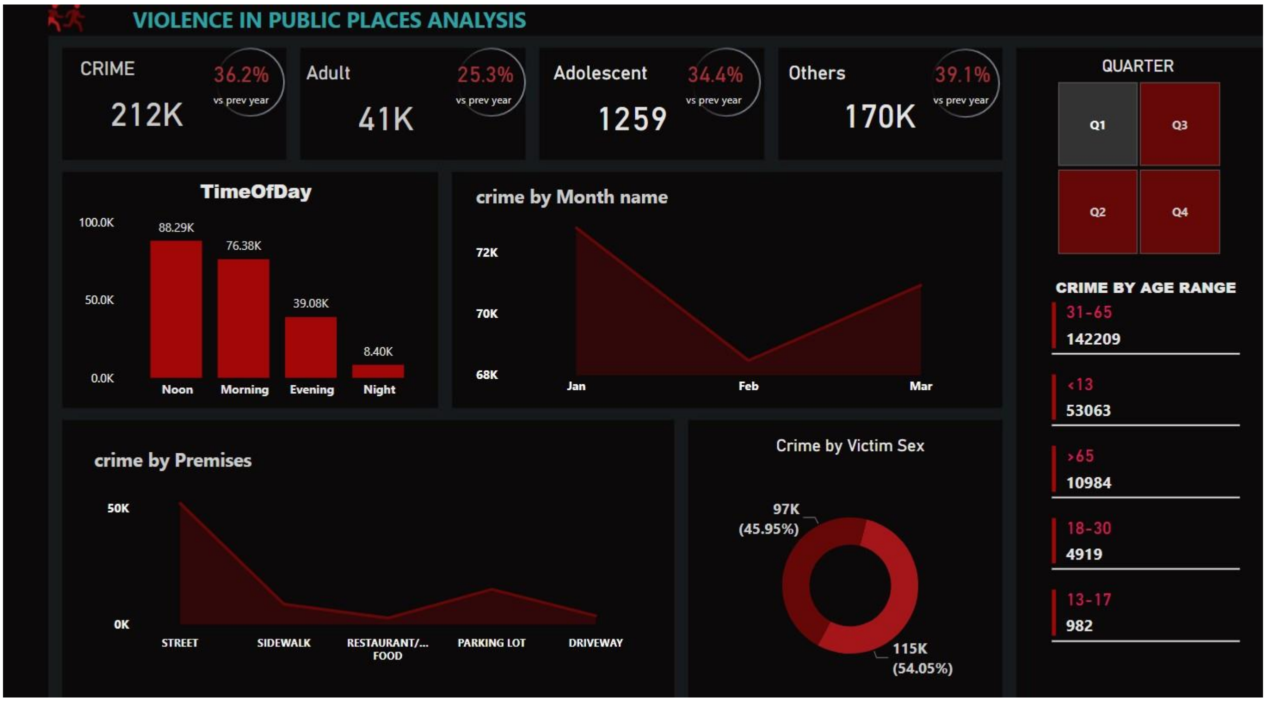


Fig A.36 Violence in Public Place in First Quarter

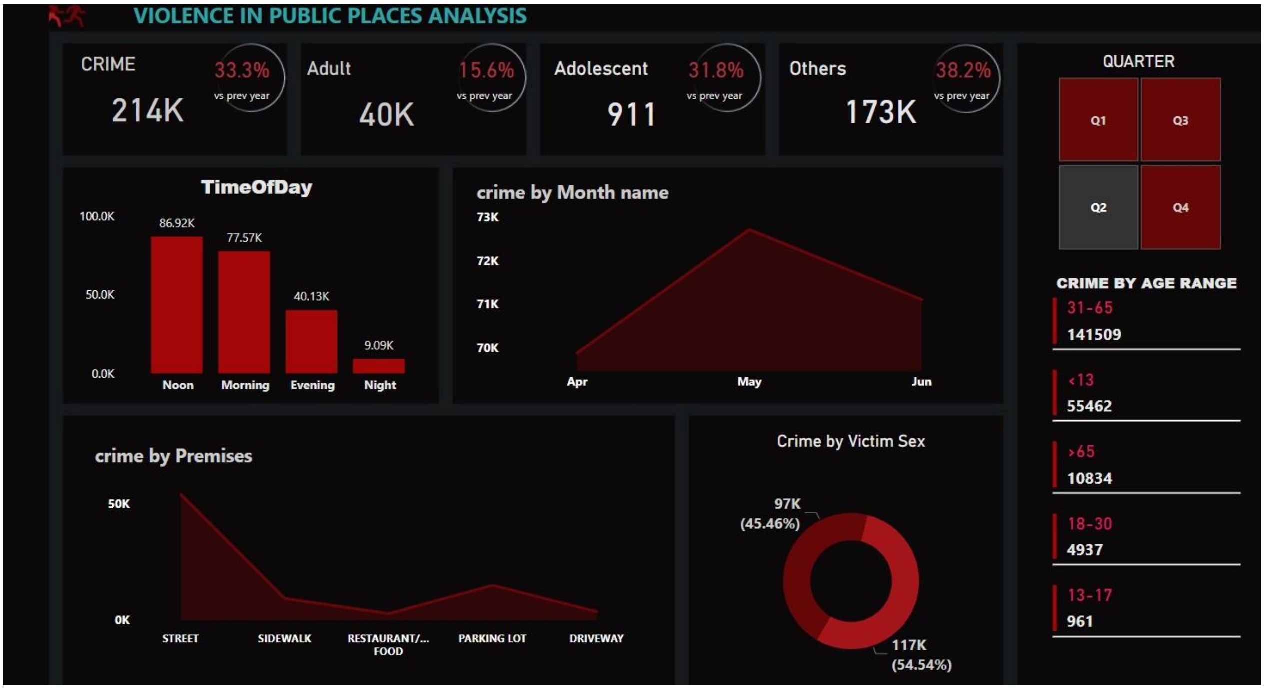


Fig A.37 Violence in Public Place in Second Quarter

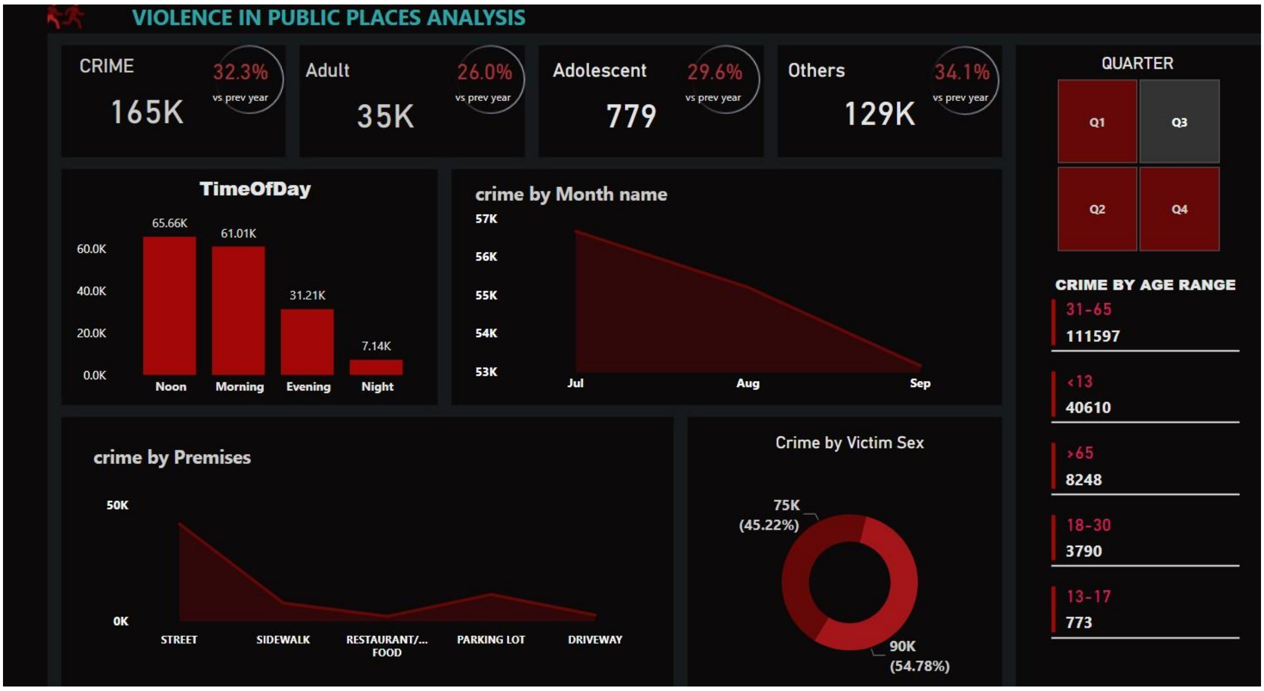


Fig A.38 Violence in Public Place in Third Quarter

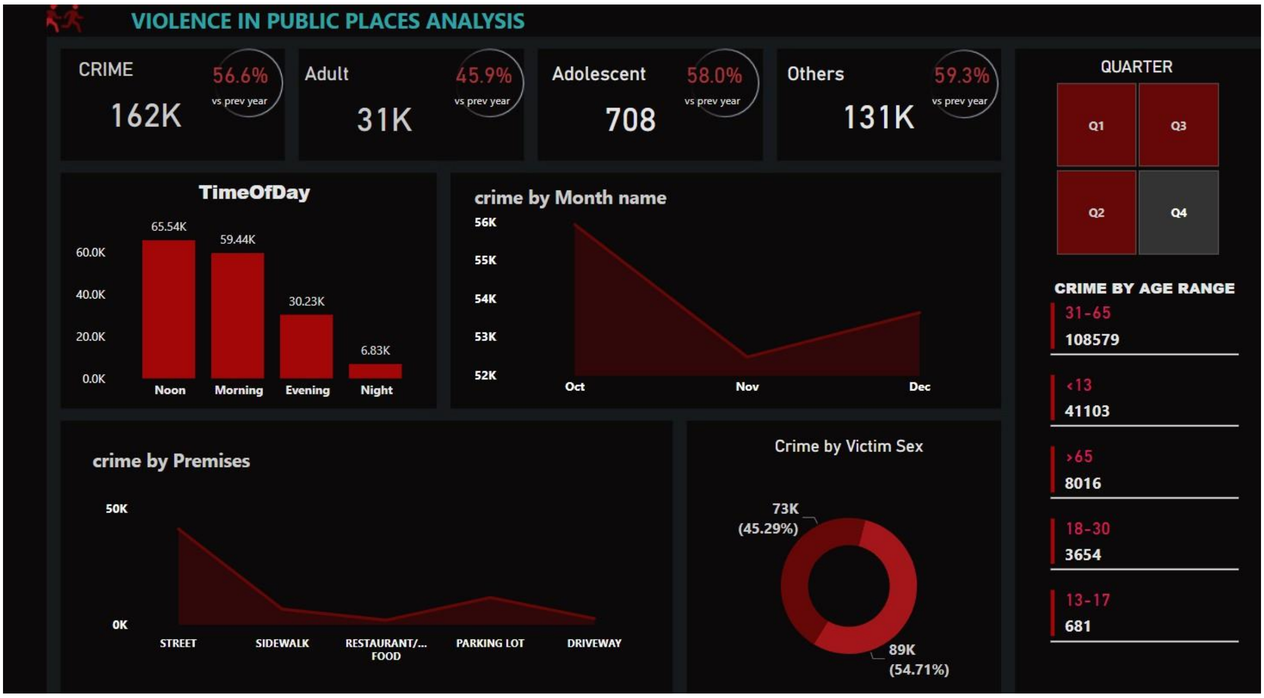


Fig A.39 Violence in Public Place in Fouth Quarter

The Report is Generated by DrillBit Plagiarism Detection Software

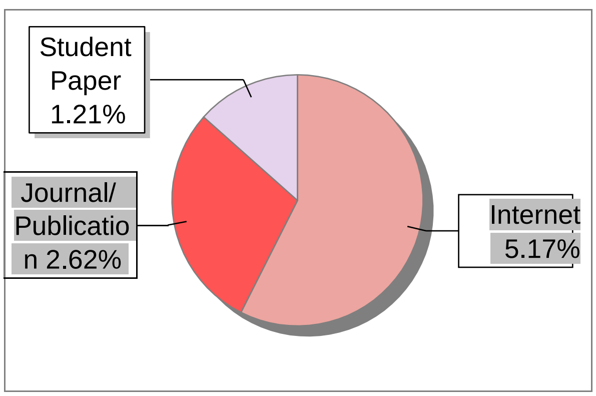
Submission Information

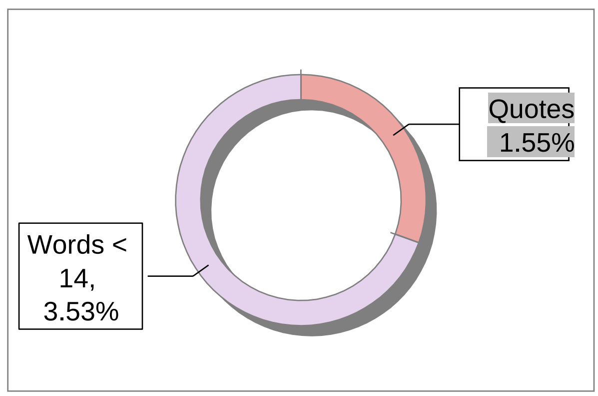
Author Name	SATHVIK S UPADHYA
Title	DESIGN OF INTELLIGENT SURVEILLANCE SYSTEM FOR MULTIPLE ANOMALY DETECTION USING DEEP LEARNING
Paper/Submission ID	1771797
Submitted by	libraryjssate@gmail.com
Submission Date	2024-05-09 10:18:28
Total Pages	82
Document type	Project Work

Result Information

Similarity **9 %**

1 10 20 30 40 50 60 70 80 90





Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words %	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

9	**46**	**A**	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	www.scribd.com	2	Internet Data
2	REPOSITORY - Submitted to P.E.S. College of Engineering Mandya on 2023-07-21 11-20	1	Student Paper
3	viso.ai	<1	Internet Data
4	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
5	www.ncbi.nlm.nih.gov	<1	Internet Data
6	etd.aau.edu.et	<1	Publication
7	Sa1997 EFTR AND STER FOR GASTROINTESTINAL SUBEPITHELIAL TUMORS (SETS) LARGE SER by Stavropoulos-2020	<1	Publication
8	cfr.annauniv.edu	<1	Publication
9	github.com	<1	Internet Data
10	www.linkedin.com	<1	Internet Data
11	dspace.nm-aist.ac.tz	<1	Publication
12	aro.koyauniversity.org	<1	Internet Data
13	citeseerx.ist.psu.edu	<1	Internet Data

14	github.com	<1	Internet Data
15	Hough Forest Random Field for Object Recognition and Segmentation by Payet-2013	<1	Publication
16	dspace.srmist.edu.in	<1	Publication
17	www.sit.ac.in	<1	Publication
18	Generative Neural Networks for Anomaly Detection in Crowded Scenes, by Wang, Tian Qiao, M- 2018	<1	Publication
20	www.liebertpub.com	<1	Internet Data
21	beei.org	<1	Publication
22	docplayer.info	<1	Internet Data
23	docplayer.net	<1	Internet Data
24	INFORMATICSwww.jatit.org	<1	Publication
25	sist.sathyabama.ac.in	<1	Publication
26	jssaherdatalake.blob.core.windows.net	<1	Publication
27	The integration of mental health services in a central New York community by Patric-1996	<1	Publication
28	www.codecademy.com	<1	Internet Data
29	clockss.org	<1	Internet Data
30	documents.mx	<1	Internet Data
31	dspace.srmist.edu.in	<1	Publication
32	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper

33	arxiv.org	<1	Internet Data
34	bbvaopen4u.com	<1	Internet Data
35	docplayer.net	<1	Internet Data
36	docplayer.net	<1	Internet Data
37	docplayer.net	<1	Internet Data
38	download.cnet.com	<1	Internet Data
39	ero-affiliates.biz	<1	Internet Data
40	ijarsct.co.in	<1	Publication
41	moam.info	<1	Internet Data
42	Municipal solid waste source-separated collection in China A comparative analys by Ju-2011	<1	Publication
43	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
44	www.doaj.org	<1	Publication
45	www.freepatentsonline.com	<1	Internet Data
46	www.indianjournals.com	<1	Publication
47	www.jpmorgan.com	<1	Internet Data