A
Mini Project Report

On

"Abnormal Event Detection on Pathway Using YOLO"

Submitted in partial fulfillment of the

Requirements for the award of the degree of

Bachelor of Technology

In

Information Technology

By

Aswini Ramachandran 217Y1A1206
Lakpathwar Sriharsh 217Y1A1252
Balast Bhuvan 217Y1A1211

Under the guidance of

Mr.Y.Apparao
Assistant Professor

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

Department of Information Technology

2024

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

Department of Information Technology

CERTIFICATE

This is to certify that the project entitled **"Abnormal Event Detection on Pathway using YOLO"** has been submitted by **Aswini Ramachandran(217Y1A1206),Lakpathwar Sriharsh(21Y1A1252),Balast Bhuvan(217Y1A1211)** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Information Technology from Jawaharlal Nehru Technological University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide **Head of the Department**

Project coordinator **External Examiner**

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

Department of Information Technology

DECLARATION

We hereby declare that the project entitled **"Abnormal Event Detection on Pathway using YOLO"** is the work done during the period from **January 2024 to May 2024** and is submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

Aswini Ramachandran **217Y1A1206**
Lakpathwar Sriharsh **217Y1A1252**
Balast Bhuvan **217Y1A1211**

MARRI LAXMAN REDDY

INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

Department of Information Technology

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express our guidance for all of them.

First of all, we would like to express our deep gratitude towards our internal guide **- Mr.Y.Apparao, Assistant Professor, Department of IT** for his support in the completion of our dissertation. We wish to express our sincere thanks to **Dr.M.Nagalakshmi,**HOD, Dept. of IT and principal **Dr. K. SRINIVAS RAO** for providing the facilities to complete the dissertation.

We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

\

Aswini Ramachandran	**217Y1A1206**
Lakpathwar Sriharsh	**217Y1A1252**
Balast Bhuvan	**217Y1A1211**

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

Department of Information Technology

ABSTRACT

To enhance road safety and prevent accidents, this project focuses on developing a system to detect abnormal activities on roads in real-time. With YOLOv8, an artificial neural network model used for object detection, our approach aims to identify various irregularities such as accidents, reckless driving, and pedestrian crossings. Through a comprehensive training process and fine-tuning of the YOLOv8 model, we adapt it to the specific requirements of road safety monitoring. At the beginning of the configuration, we discuss combining Flaskfor web application development and Cloudinary for video storage which provides the user with user authentication, video processing, and streaming. In the end, it detects the specific objects in a video stream, records segments of the video when these objects are detected, and then uploads the recorded segments to Cloudinary.

APPENDIX-1

LIST OF FIGURES

LIST OF FIGURES

APPENDIX-2

LIST OF TABLES

LIST OF TABLES

APPENDIX-3

LIST OF ABBREVIATIONS

ABBREVIATIONS

YOLO	You Only Looked Once
RSTP	Real Time Streaming Protocol
CV	Computer Vision
GAN	General Adversarial Network
CDN	Content Delivery Network
RBV	Role based video

APPENDIX-4

REFERENCES

References

[1] Riddhi Sonkar, Sadhana Rathod, Renuka Jadhav, Deepali Patil, "Crowd abnormal behavior detection using deep learning", Informatics and Mathematics Web of Conferences 32, 2020.

[2] Megha Chhirolya, Dr. Nitesh Dubey, " Abnormal Human Behavior Detection and Classification In Crowd Using Image Processing ", Conference Proceeding Issue Published in International Journal of Trend in Research and Development (IJTRD), 2013.

[3] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, Nicu Sebe," Abnormal event detection in videos using generative adversarialnets", IEEE International Conference on Image Processing,22 February 2018.

[4] Lili Cui, Kehuang Li, Jiapin Chen, Zhenbo Li," Abnormal event detection in traffic video surveillance based on local features", IEEE International Conference on Image Processing,12 December 2011.

[5] Waqas Sultani, Chen Chen, Mubarak Shah," Real-World Anomaly Detection in Surveillance Videos", IEEE/CVF Conference on Computer Vision and Pattern Recognition,16 December 2018.

[6] Louis Kratz, Ko Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models IEEE/CVF Conference on Computer Vision and Pattern Recognition, 20 June 2009.

[7] Dong-Gyu Lee, Heung-Il Suk, Sung-Kee Park, Seong-Whan Lee, Motion Influence Map for Unusual Human Activity Detection and Localization in Crowded Scenes, IEEE Transactions on Circuits and Systems for Video Technology, 28 January 2015.

[8] Kothapalli Vignesh; Gaurav Yadav; Amit Sethi, Abnormal Event Detection on BMTT-PETS 2017 Surveillance Challenge, IEEE Computer Society Conference onComputer Vision and Pattern Recognition Workshops (CVPRW), 24 August 2017.

[9] Weixin Li; Vijay Mahadevan; Nuno Vasconcelos, Anomaly Detection and Localization in Crowded Scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 JUNE 2013.

[10] Borislav Antić; Björn Ommer, Video parsing for abnormality detection, International Conference on Computer Vision (ICCV), 12 JAN 2012.

[11] Direkoglu, H., & Kilic, O. F. Abnormal Crowd Behavior Detection Using Motion Information Images and Convolutional Neural Networks. IEEE Transactions on Circuits

and Systems for Video Technology, 2017.

[12] Bokun Wang, Caiqian Yanga, Yaojing Che, Detection Anomaly based on Data Support Vector Data Description, Computational Intelligence and Neuroscience 2022.

[13] Manjula Pattnaik, Abnormal Event Detection in Pedestrian Pathway Using Garch Model and MLP Classifier, Computer Science 2019.

[14] Kyung Joo Choi, Temporal saliency-based suspicious behavior pattern detection, Applied Sciences 2020.

[15] Ali Atghaei, Soroush Ziaeinejad, Hamid Reza Alizadeh, Abnormal Event Detection in Urban Surveillance Videos Using GAN and Transfer Learning 2020.

[16] Yuxing Yang, Zeyu Fu, Syed Mohsen Naqvi Abnormal event detection for video surveillance using an enhanced two-stream fusion method, 7 October 2023.

[17] Bo Yan, Cheng Yang, Chuan Shi, Jiawei Liu, Xiaochen Wang Abnormal Event Detection via Hypergraph Contrastive Learning, 2 Apr 2023.

[18] Cewu Lu, Wei-Ming Wang, Jiaya Jia Fast Abnormal Event Detection Published online: 1 December 2018.

[19] Hoang Duy Trinh, Lorenza Giupponi, Paolo Dini Urban Anomaly Detection by processing Mobile Traffic Traces with LSTM Neural Networks 05 September 2019.

[20] Dimitriou N, Lalas A, Dasygenis M Real-Time Abnormal Event Detection for Enhanced Security in Autonomous Shuttles Mobility Infrastructures, 1 Sep 2020.

[21] Samir Bouindour, Ronghua Hu, Hichem Snoussi," Enhanced Convolutional Neural Network for Abnormal Event Detection in Video Streams", IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE),08 August 2019.

[22] Weichao Zhang, Guanjun Wang, Mengxing Huang, Hongyu Wang, Shaoping Wen," Generative Adversarial Networks for Abnormal Event Detection in Videos Based on Self-Attention Mechanism", IEEE Access Volume: 9,07 September 2021.

[23] Seongheon Park, Hanjae Kim, Minsu Kim, Dahye Kim, Kwanghoon Sohn," Normality Guided Multiple Instance Learning for Weakly Supervised Video Anomaly Detection", IEEE/CVF Winter Conference on Applications of Computer Vision (WACV),06 February 2023.

[24] Yu Zhao, Lei Zhou, Keren Fu, Jie Yang," Abnormal event detection using the spatiotemporal feature and nonnegative locality-constrained linear coding", IEEE International Conference on Image Processing (ICIP),19 August 2016.

[25] Devashree R. Patrikar, Mayur Rajaram Parate," Anomaly Detection by Predicting

Future Frames using Convolutional LSTM in Video Surveillance",2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS),02 June 2023.

[26] Xianlin Zeng, Yalong Jiang, Wenrui Ding, Hongguang Li, Yafeng Hao, Zifeng Qiu," A Hierarchical Spatio-Temporal Graph Convolutional Neural Network for Anomaly Detection in Videos", IEEE Transactions on Circuits and Systems for Video Technology Volume: 33,10 December 2021.

[27] P. Pandiaraja, Saarumathi R, Parashakthi M, Logapriya R," An Analysis of Abnormal Event Detection and Person Identification from Surveillance Cameras using Motion Vectors with Deep Learning", Second International Conference on Electronics and Renewable Systems (ICEARS),05 April 2023.

[28] Samir Bouindour, Mohamad Mazen Hittawe, Sandy Mahfouz, Hichem Snoussi," Abnormal event detection using convolutional neural networks and 1-class SVM classifier",8th International Conference on Imaging for Crime Detection and Prevention (ICDP 2017),07 June 2018.

[29] Thittaporn Ganokratanaa, Supavadee Aramvith, Nicu Sebe," Anomaly Event Detection Using Generative Adversarial Network for Surveillance Videos", Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC),05 March 2020.

[30] Hongkai Chen; Xiaoguang Zhao; Tianzheng Wang; Min Tan; Shiying Sun" Spatial-temporal context-aware abnormal event detection based on incremental sparse combination learning",12th World Congress on Intelligent Control and Automation (WCICA),29 September 2016.

INDEX

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The project focuses on implementing an Abnormal Event Detection system for roadways using YOLO for object detection. It aims to quickly recognize abnormal events like accidents, reckless driving, and pedestrian crossings to expedite emergency responses and enhance road safety. The system integrates various components like library integration, cloud storage configuration, video detection mechanism, and operational workflow to achieve real-time monitoring and analysis.The project utilizes YOLO for object detection to identify abnormalities on roads such as accidents and pedestrian crossings.It integrates library integration, cloud storage configuration, and video detection mechanisms to enable real-time monitoring and analysis.The system aims to provide timely alerts to authorities for quick responses to potential threats to street safety.

1.2 PURPOSE OF THE PROJECT

The project aims to enhance road safety by developing a system for real-time detection of abnormal activities on roads and highways through integration of event detection with CCTV cameras. By leveraging advanced computer vision techniques and deep learning models, we created a real-time system that can identify and respond to specific events of interest , such as accidents , Chain snatching , Anomaly Detection in Traffic Flow, Fighting. It focuses on using the YOLOv8 object detection model to identify irregularities such as accidents, reckless driving, and pedestrian crossings. The system is designed to provide timely alerts to authorities for quick responses to potential threats to street safety.The software permits consumer authentication, video add, live video streaming from diverse resources (uploaded documents, webcams, RTSP feeds), and offers a dashboard to manipulate recorded video clips. The foremost aim is to detect and record incidents which include chain snatching, kidnapping, injuries, and combating, with audio signals precipitated whilst these activities aredetected. The recorded video clips are stored with relevant metadata and can be accessed based totally on the user's function and permissions. The assignment combines laptop imaginative and prescient techniques, net improvement, and person authentication to create a comprehensive machine for monitoring and recording important occasions from video streams.

1.3 MOTIVATION

Through real-time road activity detection, the initiative seeks to improve road safety. to assist the public, offer protection in areas that are not authorized, and notify the police and health agencies. The system focuses on recognizing abnormalities like accidents, careless driving, and pedestrian crossings by using the YOLOv8 object identification model. The intention is to promptly notify relevant authorities so they can act upon any hazards to road safety. To facilitate quick emergency responses and prevent secondary incidents.The inclusion of person authentication and position-primarily based get entry indicates a capability application in protection or surveillance scenarios, where special ranges of get entry to recorded motion pictures can be required. The integration of audio alerts further highlights the want for set off notification while incidents arise, allowing for well timed reaction or intervention.The motivation behind our project is the development of an intelligent and practical system that can automatically detect and capture evidence of specific incidents from video streams, while offering features like user authentication, video management, and real-time alerts, making it a valuable tool for security, law enforcement, or other domains where monitoring and documenting such events is crucial.

CHAPTER 2

LITERATURE SURVEY

An extensive literature survey has been conducted by studying existing systems of Abnormal event detection on pathways using YOLO. A good number of research papers, journals, and publications have also been referred before formulating this survey.

2.1 EXISTING SYSTEM

One problem that continues to arise is the precise identification of small objects in an image. Owing to its utilization of feature maps with varying scales to encompass both large and small items, YOLOv8 may occasionally be unable to recognize smaller objects such as solitary individuals in a busy environment or little kitchen equipment. My observation is that the model has trouble identifying distinct features from small object parts, particularly in cases when there is blurring or occlusion. Greater emphasis on improving minor object detection during training could help overcome this limitation.

Accuracy issues arise when working with datasets that have a lot of different object classes. With more than 80 classes in the COCO dataset, YOLOv8's average precision is lowered due to its greater susceptibility to false positives than certain other models. The new anchor-free head could be a factor in this problem. Using distinct classifiers for each group of classes could be one way to increase differentiation. That might, however, result in a slower rate of inference. Regarding how well YOLOv8 handles data that is not part of the training set, such as new classes, scene contexts, image corruptions, etc., there are still unanswered concerns. Although generalizability was supposed to be enhanced by Cross-Iteration Batch Normalisation, more thorough testing on a variety of unlabeled datasets is required. More extensive and diverse data augmentation during real-world performance lag training could help cover more scenarios.

While YOLOv8 has come a long way, there is still more to be done to ensure that the system is resistant to new test data, prevent false positives, and handle small objects. Since it is a real-time object detection system, accuracy gains, and efficiency must be balanced.

2.2 DISADVANTAGES OF EXISTING SYSTEM

a. Manual video monitoring systems heavily rely on human operators, which can be costly, labor-intensive, and susceptible to human errors due to fatigue or distractions.

b. Human operators may not be able to continuously monitor multiple video streams simultaneously, potentially missing critical events or incidents.

c. Rule-based systems often have predefined rules or templates, which may not be able to accurately detect complex events or handle variations in lighting, occlusions, or object appearances, leading to false positives or missed events.

d. Traditional rule-based systems may struggle to adapt to changing environments, new event types, or evolving detection requirements, as they rely on manual updates or modifications to the predefined rules.

e. Manual monitoring or slow processing times in rule-based systems can lead to delayed detection and recording of events, potentially missing crucial evidence or hindering timely response and intervention.

f. Existing systems may not be designed to scale seamlessly to handle large numbers of video sources or high-resolution video streams, limiting their applicability in large-scale deployments.

g. Traditional systems may lack advanced data analysis capabilities, making it challenging to extract insights, generate reports, or identify patterns from recorded events and videos.

2.3 DIFFERENCE BETWEEN RTSP FEED AND LIVE WEB CAM

FEATURE	RTSP FEED	LIVE WEBCAM
Protocol	RTSP (Real-Time Streaming Protocol)	Typically HTTP or other web-based streaming protocols
Usage	Often used for IP cameras, network surveillance, and media servers	Commonly used for video calls, live streaming on social media, and video conferencing
Streaming Method	RTSP for control, RTP (Real-Time Transport Protocol) for data	Usually HTTP/HTTPS for web streaming, sometimes RTMP for live streaming platforms
Latency	Generally low latency, suitable for real-time applications	Latency varies; can be higher depending on the platform and network conditions
Configuration	Requires RTSP-compatible server and client; more complex setup	Typically easier to set up with built-in or plug-in support for various applications
Quality	Allows for high-quality streams with adjustable settings	Quality varies based on the webcam and streaming platform capabilities
Control Features	Supports pause, resume, seek, and other control features	Limited control features; primarily focused on live streaming without advanced controls

CHAPTER 3

PROPOSED SYSTEM

3.1 PROPOSED SYSTEM

The proposed device is an actual-time video event detection and recording device that utilizes deep mastering and computer imaginative and prescient strategies to constantly display video streams and routinely discover and record unique events of hobby, which include chain snatching, kidnapping, injuries, and prevention. It accepts video input from numerous sources like IP cameras, CCTV systems, and pre-recorded documents.

At its core is the YOLO (You Only Looked Once) item detection version, a latest deep studying algorithm skilled on a dataset precise to the occasions of hobby, capable of detecting and classifying more than one objects simultaneously with excessive accuracy and pace. When an event is detected, the gadget triggers the recording mechanism, shooting video frames before, throughout, and after the occasion, creating a comprehensive video clip stored to a chosen garage place with appropriate naming and metadata.

Alerts and notifications may be generated and added thru diverse channels to relevant authorities or monitoring employees upon occasion detection. The system presents a person-pleasant interface for monitoring stay video, gaining access to recorded clips, and filtering events based totally on standards like kind, place, and time, with real-time item detection effects overlaid. It carries sturdy storage and database answers for handling recorded clips and associated metadata, both regionally or cloud-based.

3.2 OBJECTIVES OF PROPOSED SYSTEM

1. Develop a system that automatically analyzes uploaded videos to identify unusual occurrences.
2. Utilize OpenCV, a computer vision library, to process video data, including object detection (using YOLO) and feature extraction to understand scene content.
3. Compare identified objects and movements to established patterns of normal behavior, flagging deviations as potential anomalies.

4. Design a user-friendly web interface for uploading videos, interacting with the system, and visualizing detected anomalies (e.g., timestamps or highlighted sections).
5. Integrate cloud storage to ensure scalable storage of uploaded videos and analysis results for later retrieval or visualization.

3.3 ADVANTAGES OF PROPOSED SYSTEM

The proposed system has the following advantages:

a. The system leverages advanced deep learning techniques to automatically detect and classify events of interest, such as chain snatching, kidnapping, accidents, and fighting, without the need for constant human monitoring.
b. The YOLO object detection model used in the system is highly accurate and reliable, minimizing false positives and ensuring that critical events are not missed.
c. The system captures video frames before, during, and after an event, providing a complete context and evidence for analysis and investigation.
d. The recorded video clips are automatically stored with appropriate naming conventions and metadata, enabling efficient retrieval and organization for future reference or investigation.
e. The modular architecture of the system allows for easy integration with existing security systems, scalability to handle multiple video sources, and adaptability to evolving requirements or detection needs.
f. The system can generate alerts and notifications upon event detection, enabling remote monitoring and immediate notification to relevant authorities or personnel.
g. The system provides a user-friendly interface for monitoring live video streams, accessing recorded clips, and filtering events based on various criteria, enhancing usability and streamlining the analysis process.
h. Cost-Effective Solution: Leveraging advanced computer vision and deep learning techniques, the system can provide a cost-effective solution for event detection and monitoring compared to traditional methods that rely heavily on human resources.

3.4 SYSTEM REQUIREMENTS

The system requirements for the development and deployment of the project as an application are specified in this section. These requirements are not to be confused with the end-user system requirements. There are no specific, end-user requirements as the intended application is web based feedback form and is supposed to work on devices of all configurations.

3.4.1 SOFTWARE REQUIREMENTS

Below are the software requirements for application development:

a. Editor for HTML, CSS, Python, Streamlit, Flask - VS Code
b. Opencv
c. Cloudinary SDK for store video.
d. Google collab and jupiter notebooks for training.

3.4.2 HARDWARE REQUIREMENTS

Hardware requirements for application development are as follows:

a. CPU - intel i3 or higher
b. RAM - 4 GB or higher
c. Compatible camera setup
d. Adequate storage space

3.4.3 IMPLEMENTATION TECHNOLOGIES

YOLO:

YOLO (You Only Looked Once) provided a breakthrough method that completely changed object detection.YOLO uses a single, potent neural network. This network can simultaneously predict object classes and bounding boxes for every item in a picture, in a "one shot." This novel single-shot technique is made possible by YOLO's distinct fully connected layer, which also makes it substantially quicker than current algorithms like R-CNN. For instance, faster R-CNN has a slower processing time since it needs moreiterations to find possible objects and then classify them.

Flask:

Flask is a Python-made, tiny website coding tool. It's great since it's tiny and can do lots of things. Developers use Flask to build webs fast without much code prep. Flask follows a "micro" way. It gives tools for web things like e-mail, URL control, display help, and word tracking but not too much. Flask doesn't have built-in data trackers or word checkers. Yet, coders can add special plug-ins to Flask for those parts. These extra Flask add-ons handle lots of web jobs. They track data links, validate form inputs, transfer files, authorize users, and more common web tasks. Its simplicity and excellent documentation make it popular for developers creating web apps, APIs, and microservices.

Web Technologies:

Our abnormal event detection system utilizes web technologies for both the user interface and processing. HTML and CSS build the user interface for interaction, while Python with Flask forms the core for web application functionalities and communication with OpenCV, a computer vision library that analyzes uploaded videos. Cloudinary SDK interacts with cloud storage for efficient video management. This combination of web technologies allows usersto interact with the system and leverage its video analysis capabilities.

Cloudinary:

Cloudinary streamlines image and video management in the cloud. Offering secure storage accessible from anywhere.Cloudinary acts as a media management powerhouse. It automatically optimizes your content for various devices and web speeds, ensuring fastloading times and a smooth user experience. Additionally, Cloudinary utilizes a Content Delivery Network (CDN) for efficient global delivery. No matter the user's location, content loads quickly. The real power lies in on-the-fly transformations. Cloudinary also handles itall in real time, eliminating the need for manual editing.

Opencv:

OpenCV (Open Source Computer Vision Library) is a powerful open-source software library designed for computer vision and machine learning applications. It provides a comprehensive suite of tools and algorithms for image and video processing, enabling developers to perform tasks such as image manipulation, feature detection, object recognition, and video analysis. With support for various programming languages including C++, Python, and Java, OpenCV is widely used in fields like robotics, surveillance, medical imaging, augmented reality, and autonomous vehicles, making it a versatile and essential tool for developers working in computer vision and AI domains.

CHAPTER 4

SYSTEM DESIGN

4.1 PROPOSED SYSTEM ARCHITECTURE

The proposed abnormal event detection system leverages OpenCV, a powerful library, for analyzing uploaded videos. YOLO, an object detection algorithm within OpenCV, helps identify objects and their movements.

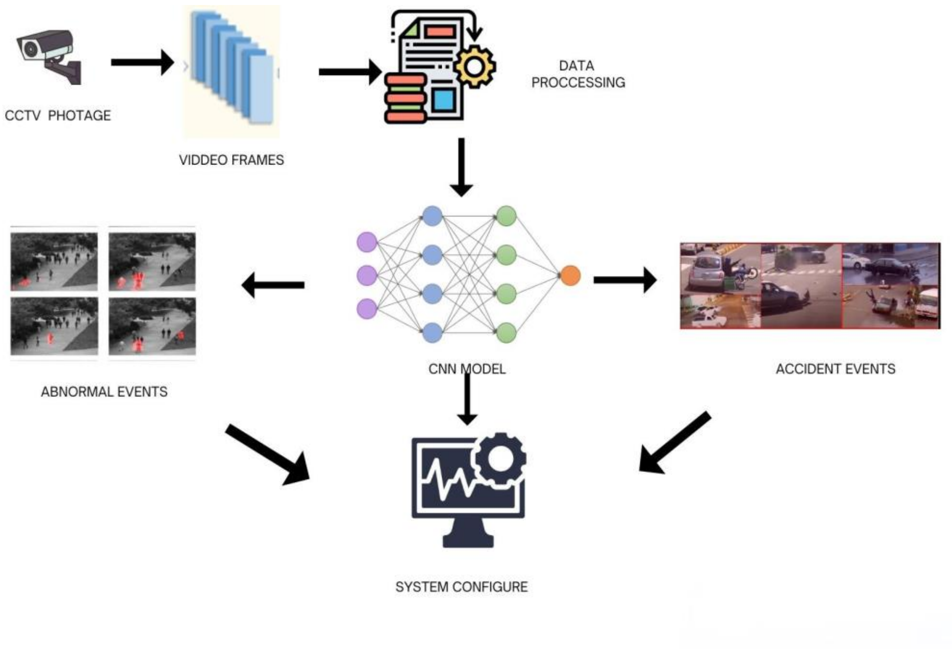

Figure 1: Architecture of Proposed solution

By comparing these observations to established patterns, the system flags unusual occurrences as potential anomalies. Cloud storage ensures efficient video management, and the user-friendly interface allows easy interaction and visualization of these detected anomalies.

4.2 UML DIAGRAMS

The UML Diagrams overall involve three main diagrams-.

4.2.1 USE CASE DIAGRAM:

This use case diagram illustrates the everyday drift of movements for an atypical occasion detection machine. After logging in, a person can carry out three major obligations: view the dashboard, manipulate customers, and deal with video-related operations.

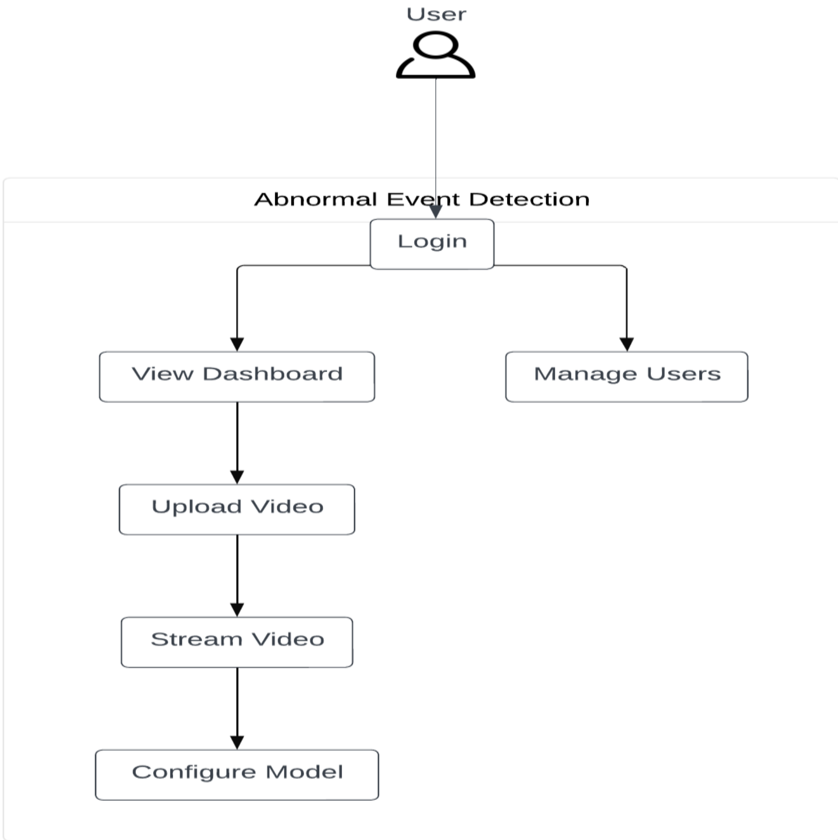

Figure 2: Use case diagram of the proposed system

The "View Dashboard" functionality lets the consumer get admission to and reveal the device's dashboard, which probably displays applicable facts and analytics associated with odd event detection.

The "Manage Users" alternative enables the person to feature, alter, or eliminate consumer debts inside the system, controlling admission to and permissions.

The video-related operations consist of 3 steps: importing a video, streaming the video, and configuring the model used for atypical occasion detection. The person first uploads a video document, then streams or performs the video, and ultimately configures the underlying version or algorithm answerable for detecting extraordinary occasions in the video photos.

This use case diagram gives an immoderate-level evaluation of the device's center functionalities, showcasing the logical float of actions a person should observe to correctly make use of the odd occasion detection device.

4.2.2 ACTIVITY DIAGRAM

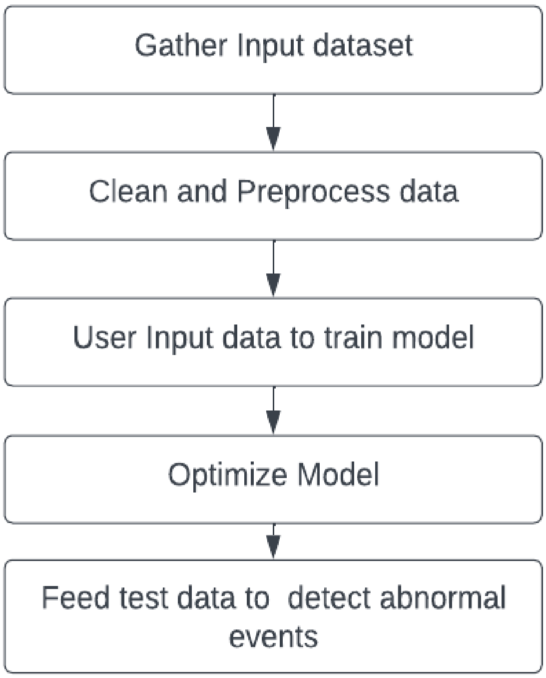

Figure 3: Activity diagram of the proposed system

The technique starts off evolving by amassing the necessary input dataset, which could encompass video pictures, pics, or other relevant statistics resources for schooling the ordinary occasion detection model. Once the input dataset is collected, it undergoes a cleansing and preprocessing step to ensure statistics first-class and consistency. This might also involve tasks along with noise removal, facts formatting, and different records practice sports. The wiped clean and preprocessed data is then used as input to educate the bizarre occasion detection model, regarding feeding the organized facts into the selected device studying algorithm or model to study styles and characteristics related to every day and bizarre occasions. During the schooling technique, the version's basic typical performance is evaluated, and diverse optimization strategies are implemented to enhance its accuracy and ordinary overall performance in detecting everyday activities. Finally, as soon as the model iseducated and optimized, it could be deployed to detect atypical occasions in new input statistics. This very last step involves taking a look at statistics or live facts sources, such as video streams, to the trained model, which then analyzes the records and identifies any extraordinary events or styles.

4.2.3 SEQUENCE DIAGRAM:

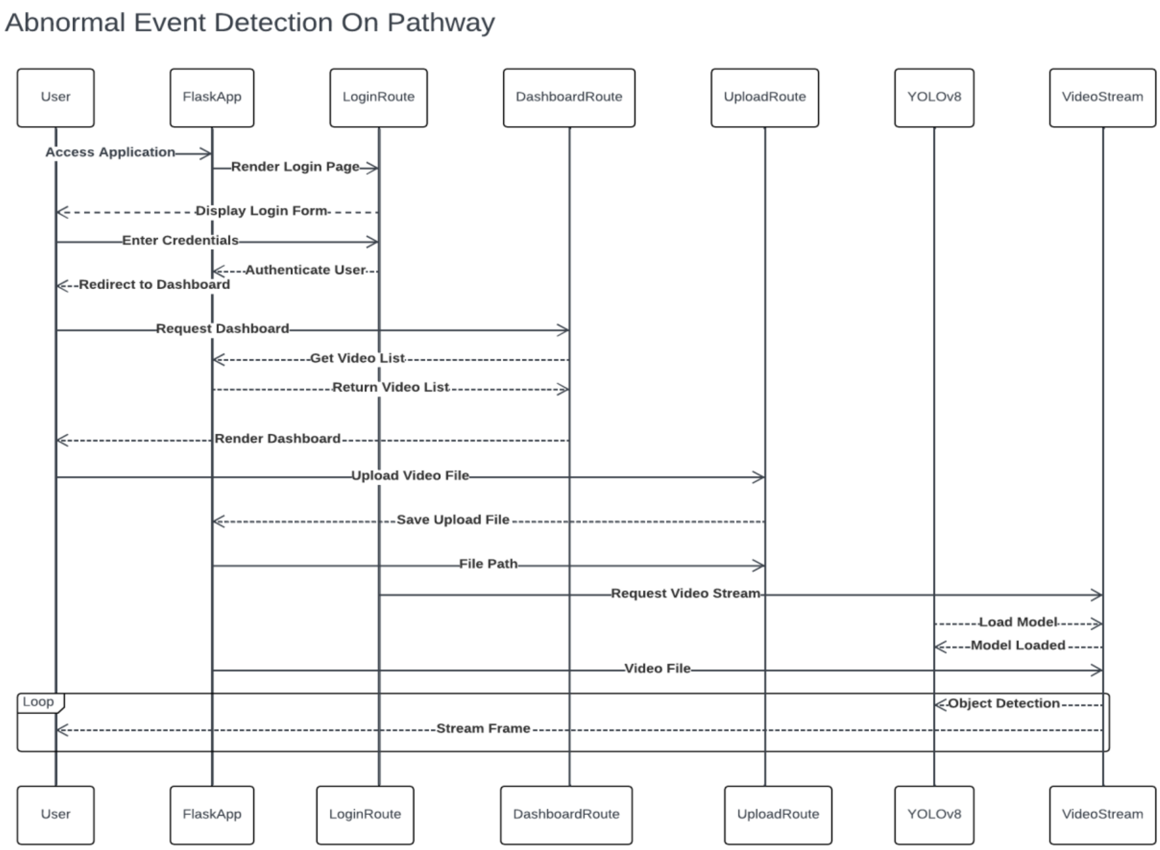

Figure 4: Sequence diagram of the proposed system

The given image is a sequence case diagram that illustrates the interaction among exceptional components of a peculiar occasion detection device, in particular focusing on the pathway or going with the flow of moves.

The person first accesses the utility, which renders the login web page through the FlaskApp and LoginRoute components. The person then enters their credentials, which can be authenticated. Upon a hit authentication, the consumer is redirected to the dashboard.

The DashboardRoute factor requests the video list, that is retrieved and lower back, allowing the dashboard to render with the listing of available films.

When the user uploads a video file, the UploadRoute aspect saves the uploaded report and passes the report route to the YOLOv8 factor. This issue is possibly liable for loading the necessary model or algorithm used for item detection.

Once the version is loaded, the consumer can request a video stream. The VideoStream component then streams the video body by way of the body to the YOLOv8 factor, which performs object detection on every frame. The series diagram illustrates the collection of interactions among the various components, including the consumer interface (FlaskApp), routing additives (LoginRoute, DashboardRoute, UploadRoute), item detection version (YOLOv8), and video streaming (VideoStream).

4.2.4 CLASS DIAGRAM:

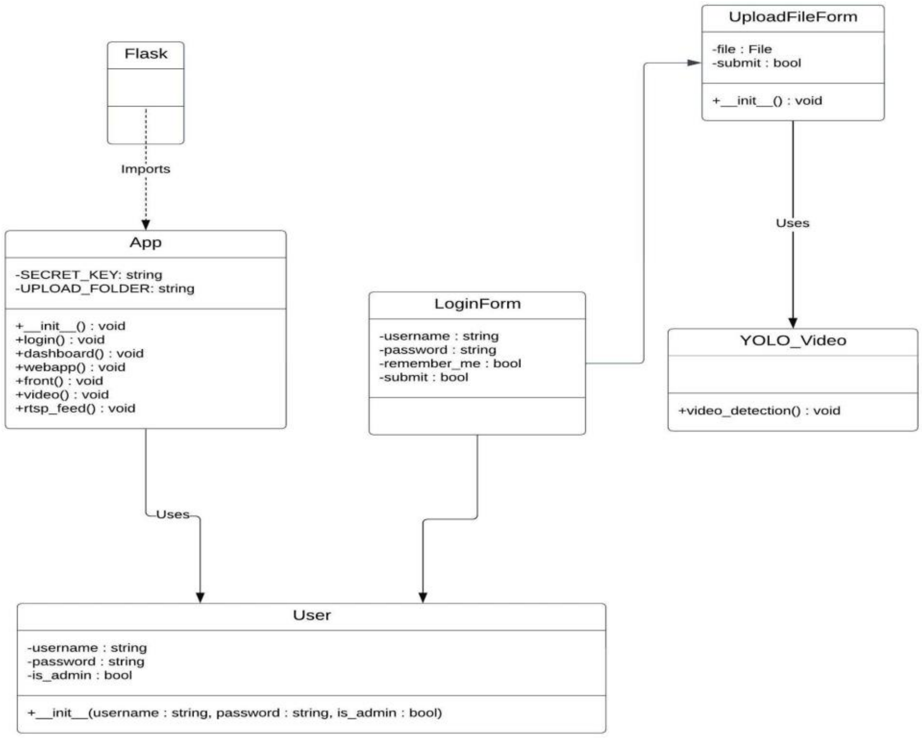

Fig 5. Class diagram of the proposed system

In this diagram, the Flask program is imported by using the App magnificence, which serves as the main application magnificence. The App elegance has techniques for coping with different routes and functionalities, together with login, dashboard, video streaming, and others.The User elegance represents a consumer with houses like username, password, and is_admin. This class is in all likelihood used for authentication purposes.The LoginForm and UploadFileForm instructions are Flask-WTF bureaucracy used for dealing with consumer input for login and record upload, respectively.The UploadFileForm elegance makes use of the video_detection() function from the YOLO_Video.Py module for item detection on the uploaded video.

CHAPTER 5

IMPLEMENTATION

5.1 IMPLEMENTATION BASED ON OBJECT DETECTION

Implementing a real-time event detection system using YOLOv8 for object detection, Flask for backend coordination, Cloudinary for scalable video storage, and developing user interfaces for hospitals and police to access relevant footage.

1. Data Collection and Data Preprocessing:

Gathering Diverse Dataset: We embark on collecting a comprehensive dataset comprising a wide array of CCTV footage capturing various abnormal events such as accidents, chain snatching, fighting, and kidnapping. This dataset is pivotal in ensuring the robustness and generalization capabilities of our model across different scenarios.

Annotating Dataset: Each video sequence in the collected dataset undergoes meticulous annotation, wherein regions corresponding to the occurrence of abnormal events are precisely marked. This annotation process is crucial for training our detection model with ground truth labels, enabling it to accurately identify and localize these events during inference.

Data Preprocessing: To enhance the model's resilience and performance, we meticulously preprocess the annotated dataset. This involves resizing video frames to a consistent resolution, normalizing pixel values to a standardized range, and applying data augmentation techniques such as rotation, scaling, and flipping. These preprocessing steps aid in mitigating overfitting and equipping the model with robust feature representations for effective event detection.

2. Implementing the YOLOv8 Model:

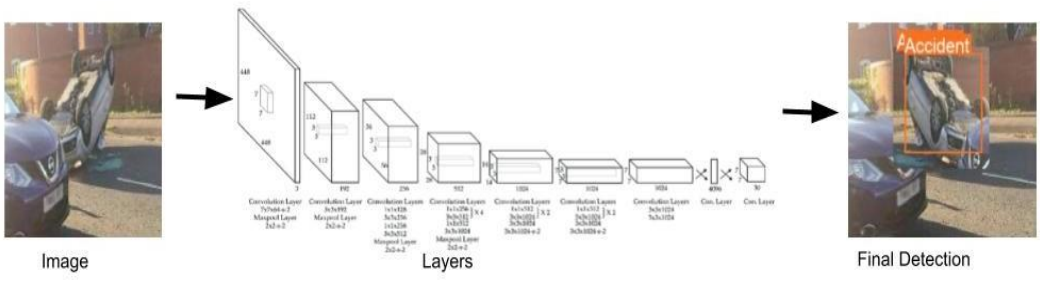

Fig 6. Detecting the Objects using YOLOv8 Algorithm

Utilizing Pre-trained Weights: Leveraging the power of transfer learning, we kickstart our implementation by initializing the YOLOv8 model with pre-trained weights. These weights, acquired from training on large-scale datasets, provide a strong foundation for detecting a diverse range of objects, including those relevant to our abnormal event detection task.

Fine-tuning on Annotated Dataset: Building upon the pre-trained weights, we fine-tuned the YOLOv8 model using our annotated dataset. Through this process, the model learns to adapt its feature representations to the nuances and intricacies of abnormal event detection, thereby enhancing its specificity and sensitivity towards recognizing such events in real-world scenarios.

Integration with Flask Backend: Seamless integration of the YOLOv8 model into our Flask backend facilitates real-time inference capabilities. By embedding the model within the backend infrastructure, we enable efficient processing of video streams from CCTV cameras, allowing for timely detection and response to abnormal events as they unfold.

3. Developing and Training the Model:

Designing Flask Backend Architecture: The architecture of our Flask backend is meticulously crafted to accommodate the reception and processing of video streams originating from CCTV cameras. This architecture encompasses robust handling of concurrent requests, ensuring seamless interaction with the YOLOv8 detection model for event identification.

Implementing Event Detection Logic: Within the Flask backend, sophisticated logic is implemented to trigger event detection mechanisms upon the identification of abnormal activities within the received video streams. This logic orchestrates the seamless integrationof the YOLOv8 model, orchestrating the detection process with precision and efficiency.

Training YOLOv8 Model: The training phase involves feeding our annotated dataset into the YOLOv8 model, optimizing its parameters to maximize performance metrics such as precision, recall, and F1-score. Through iterative training iterations, the model gradually refines its ability to discern abnormal events from background noise, culminating in a highly proficient detection system.

4. Testing the Model:

Setting Up Testing Environment: A comprehensive testing environment is established, comprising simulated CCTV feeds and pre-recorded events spanning various abnormal scenarios. This environment allows for meticulous evaluation of the YOLOv8 model's performance across diverse conditions, facilitating robustness validation and performance assessment.

Performance Evaluation Metrics: Key performance metrics such as precision, recall, and F1-score are meticulously measured and analyzed to gauge the effectiveness of the YOLOv8 model in detecting abnormal events. Rigorous testing under real-world conditions further validates the model's efficacy, ensuring its reliability and responsiveness in practical deployment scenarios.

5.Data Storing:

Integration with Cloudinary: Seamless integration with Cloudinary's API facilitates the seamless uploading of detected event segments to the cloud-based storage platform. This integration enhances data management capabilities, enabling secure and scalable storage of relevant video footage for subsequent retrieval and analysis.

User Interface Development: Tailored user interfaces are meticulously designed for hospitals and police authorities, providing intuitive access to relevant video footage based on detected events. These interfaces empower users to swiftly review and analyze pertinent information, facilitating timely decision-making and response coordination.

Security and Access Control: Robust authentication and authorization mechanisms are implemented to safeguard sensitive video data, ensuring that only authorized users, such as hospital staff and law enforcement personnel, can access and retrieve relevant footage. This stringent security framework enhances data privacy and confidentiality, bolstering user trust and compliance.

Scalability and Performance Optimization: Ongoing efforts are dedicated to optimizing system components for scalability and performance, enabling seamless handling of large volumes of video data. Through continuous refinement and optimization, our system ensures efficient processing and analysis of CCTV footage, even amidst increasing data volumes and user demands.

5.2 SOURCE CODE

Flask Application Setup

Import required libraries and modules

Define Flask app and configure it

INITIALIZE Flask app

SET app secret

SET upload folder path

Configure Cloudinary

Define User class (if using a database)

CLASS User:

DEFINE constructor(init) with username, password,

and is_admin

Login form

Define Login form

CLASS LoginForm(FlaskForm):

DEFINE form fields (username, password, remember_me,submit)

Login Route with Authentication

Define login route handler

ROUTE / (GET, POST):

IF the form is valid:

AUTHENTICATE user with stored

usernames and password

IF authentication successful:

SET session login_type based on username

REDIRECT to dashboard route

ELSE:

RENDER login template with error

ELSE:

RENDER login template

Video Filtering

Define video filtering function

FUNCTION filter_videos(videos):

DEFINE keywords

IF session login_type is 'accident_videos':

FILTER videos based on keywords

RETURN filtered videos

User File Input

Define UploadFileForm

CLASS UploadFileForm(FlaskForm):
DEFINE form fields (file, submit)

Video Frame Generation

Define frame generation functions
FUNCTION generate_frames(path_x):
CALL video_detection function with path_x

FOR each detection in output:
ENCODE detection as JPEG
YIELD frame

Dashboard Data Retrieval

Define dashboard route handler
ROUTE /dashboard (GET):
IF user not logged in:
RENDER login template with error
FETCH videos from Cloudinary
FILTER videos based on login_type
RENDER dashboard template with videos

Video Processing and Object Detection Using YOLO

```
INITIALIZE necessary variables and constants
CONFIGURE Cloudinary
CREATE directory "saved_videos" for storing  recorded videos

LOAD pre-trained YOLO object detection model
GET current time, date, and location
INITIALIZE video capture from the specified path

IF video capture failed
  RAISE ValueError
GET frame dimensions
SET buffer size and number of frames to record
WHILE video is not ended
  READ frame from video stream

  IF end of video reached
    SAVE the recorded video clip to file
    UPLOAD the video clip to Cloudinary
    BREAK the loop
  COPY the current frame for recording
  PERFORM object detection using YOLO model
  FOR each detected object
```

```
        GET bounding box coordinates and
        confidence score

        IF confidence score is above threshold
            DRAW bounding box and label on the frame
            IF recording is not in progress
                PRINT message indicating event
                detection
                START recording by setting recording
                flag
                SET start frame for recording
                RESET frames since last save counter

            ELSE IF(recording is in progress and
            desired number of frames have been
            recorded)
                STOP recording by resetting recording
                flag
                SAVE recorded video clip to
                file
                UPLOAD the video clip to Cloudinary
                RESET recording buffer and frame
                counters

    IF recording is in progress
        APPEND the current frame to the recording
        buffer
        MAINTAIN the buffer size by removing older
        frames
        if needed
        INCREMENT frames since last save counter
        IF (desired number of frames have been
        recorded)
            STOP recording by resetting recording
            flag
            RESET frames since last save counter

    DISPLAY the processed frame
HANDLE any exceptions that occurred
RELEASE video capture and close windows
PRINT 'Video processing complete'
CLEAR recording buffer
CLEAR frames_since_save counter
CLEAR video_count counter
```

CLEAR detections list

Dashboard.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Video Dashboard</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #1A1E23;
    }
        header.feature-box.right{
        background-color:#23272F;
        color:white;
        border-bottom-color: rgba(246,247,249,.05);
        border-width: 1px;
        padding:5px;
        text-align:right;
        }
        .header-ul{
        list-style:none;
        padding:0px;
        margin: 0px;
        }

    .header-li{
        display:inline-block;
        border-radius:10px;
        color:#707377;
        padding:10px;
        }
        a:hover{
        color: white;
        }
        header a{
        color:#707377;
        text-decoration:none;
        width:100%;
        }
        .details{
```

```
    height:80px;
    margin:40px;
    padding:0px;
    font-size:40px;
    background-color:gray;
    text-align:center;
    color:#1A1E23;
    }
     .container {
  max-width: 90vw;
  margin: 20px auto;
  padding: 20px;
  background-color: #fff;
  border-radius: 5px;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  background: #23272F;
}
h1 {
  text-align: center;
  margin-bottom: 20px;
  color: white;
  font-size: 36px;
  text-shadow: 2px 2px 4px rgba(0,0,0,0.1);
}
.video-container {
}
video {
  width: 100%;
  border-radius: 5px;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  transition: transform 0.3s;
  border: 2px solid #ccc;
}
video:hover {
  transform: translateY(-5px);
  filter: brightness(1.1);
}
.video-item {
  display: flex;
  width: 20rem;
  flex-direction:  column;
  background:#1A1E23;
  align-items: center;
  justify-content: space-between;
```

```
      margin-bottom: 20px;
      margin-left: 20px;
      border-radius: 5px;
      box-shadow: 0 2px 4px rgba(0,0,0,0.05);
    }
    .video-info {
      flex-grow: 1;
      padding: 10px 0px;
      background-color: #1A1E23;
      color: white;
      border-radius: 0 5px 5px 0;
    }
    .video-title {
      margin-bottom: 5px;
      font-size: 18px;
      color: white;
    }
    .video-ul{
   display:flex;
   gap:2;
   align-items: center;
   justify-content: center;
   flex-wrap: wrap;
    }
    .video-description {
      color: #666;
      font-size: 14px;
    }
  </style>
</head>
<body>
    <header class="feature-box right">
        <nav>
            <ul class="header-ul">
            <li class="header-li"><a href="/">Home</a></li>
            <li class="header-li" ><a href="/FrontPage">Video</a></li>
            <li class="header-li"><a href="/webcam">LiveWebcam</a></li>
            <li class="header-li" ><a href="/rtsp_feed">RTSP Feed</a></li>
            </ul>
        </nav>
    </header>
  <div class="container">
    <h1>Emergency Dashboard</h1>
    <div class="video-container">
```

```
      <ul class="video-ul">
        {% for video in videos %}
          {% set video_name_parts = video.split('_') %}
          {% set video_title = video_name_parts[3] %}
          {% set video_date = video_name_parts[3] %}
          {% set video_time = video_name_parts[4] %}
          {% set video_location = video_name_parts[5].split('.')[0] %}
          <li class="video-item">
            <video controls muted preload="metadata">
            <source src="{{ video }}" type="video/mp4">
            Your browser does not support the video tag.
          </video>
            <div class="video-info">
              <h2 class="video-title">{{ video.split('/')[-1].split('_')[0] }}</h2>
              <p class="video-description">
                                                  Date:    {{    video_date[8:10]    }}-{{    video_date[5:7]    }}-{{    video_date[0:4]
}}<br>
                Time: {{ video_time.replace('-', ':') }}<br>
                Location: {{ video_location }}
              </p>
            </div>
          </li>
        {% endfor %}
      </ul>
    </div>
  </div>
</body>
</html>
```

login.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background:#1A1E23;
            height: 100vh;
        }
        header.feature-box.right{
```

```
background-color:#23272F;
color:white;
border-bottom-color: rgba(246,247,249,.05);
border-width: 1px;
      padding:5px;
text-align:right;
}
ul{
list-style:none;
padding:0px;
margin: 0px;
}
li{
display:inline-block;
     border-radius:10px;
color:#707377;
padding:10px;
}
a:hover{
color: white;
}
header a{
color:#707377;
text-decoration:none;
width:100%;
}
.details{
height:80px;
margin:40px;
padding:0px;
font-size:40px;
background-color:gray;
text-align:center;
color:black;
}
.container {
    border: solid #717377 2px;
    background-color:#23272F; /* White background color for the form */
    border-radius: 8px;
    box-shadow: 0 2px 4px #717377;
    display: flex;
    height:25rem;
    justify-content: center;
    align-items: center;
```

```
    flex-direction: column;
}
.cont{
    height: 95vh;
    display: flex;
    justify-content: center;
    align-items: center;
}
h3 {
   color: white;
    text-align: center;
    margin-top: 4px;
}
form {
   height: 35vh;
   display: flex;
   min-width: 25rem;
   justify-content: center;
}
.user_input {
   display: block;
   margin-bottom: 5px;
   font-size: medium;
   margin-left: 5rem;
   color: #dcdcdc;
}
input[type="text"],
input[type="password"] {
    width: 20rem;
    color:white;
    padding: 8px;
    margin-left: 5rem;
    margin-right: 5rem;
    margin-bottom: 1.2rem;
    box-shadow: 0 1.5px 2px #717377;
    background: #23272F;
    border: 2px solid #ccc;
    border-radius: 3px;
}
input[type="checkbox"] {
    margin-right: 5px;
    display: inline-block;
    vertical-align: middle;
}
```

```
        .remember-me  {
            display: inline-block;
            vertical-align: middle;
        }
          .user_input {
          display: block;
          margin-bottom: 5px;
          font-size: medium;
          margin-left: 5rem;
          color: #dcdcdc;
       }
    </style>
</head>
<body>
    <header class="feature-box right">
        <nav>
            <ul>
            <li><a href="/">Home</a></li>
            <li><a href="/FrontPage">Video</a></li>
            <li><a href="/webcam">LiveWebcam</a></li>
            <li><a href="/rtsp_feed">RTSP Feed</a></li>
            </ul>
        </nav>
    </header>
    <div class="cont">
    <div class="container">
        <p class="text-tag ">WELCOME BACK</p>
        <h3>Log in to your account</h3>
        <form method="POST">
            {{ form.hidden_tag() }}
            <label for="username" class="user_input">{{ form.username.label }}</label>
            {{ form.username }}
            <label for="password" class="user_input" >{{ form.password.label }}</label>
            {{ form.password }}
            <!-- <label class="remember-me" for="remember_me">{{ form.remember_me }} Remember Me</label> -->
            <button type="submit">{{ form.submit.label }}</button>
        </form>
        {% if error %}
            <p class="error">{{ error }}</p>
        {% endif %}
    </div>
</div>
</body></html>
```

CHAPTER 6

RESULTS

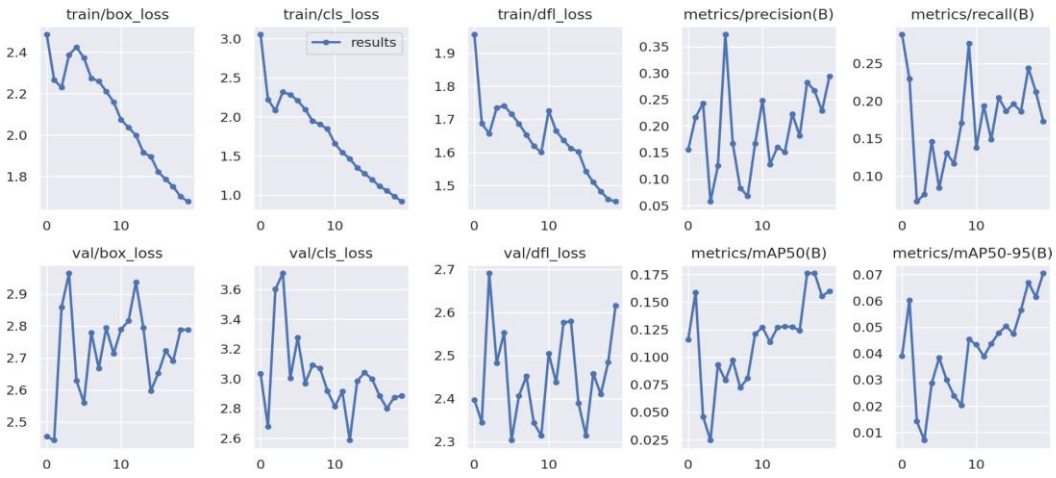

Fig 7 :Resulting Graph of the Execution

Figure 7 contains multiple line graphs plotted together, Each graph displays the values of a different metric or variable over some period.

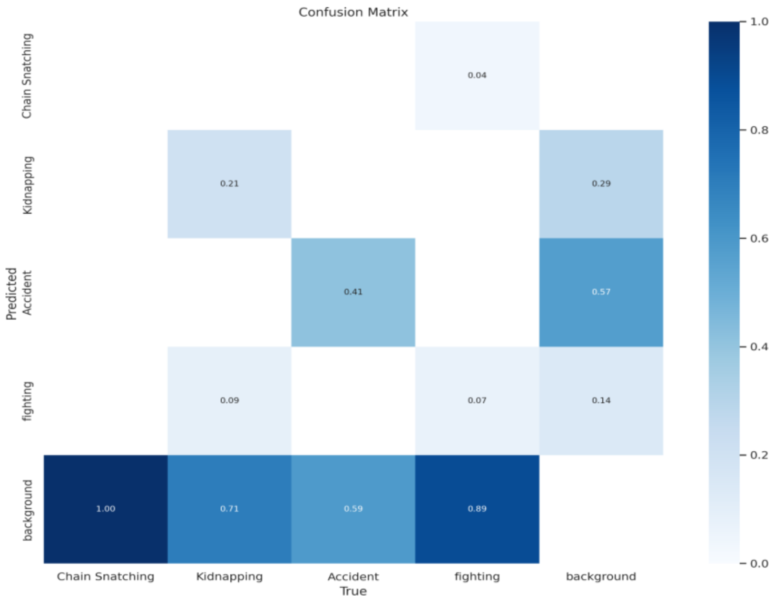

Figure 8 : confusion matrix

In Figure 8 the rows represent the ground truth labels of the images, and the columns represent the labels predicted by the model. The numbers in the cells represent the number of images that were classified correctly or incorrectly.

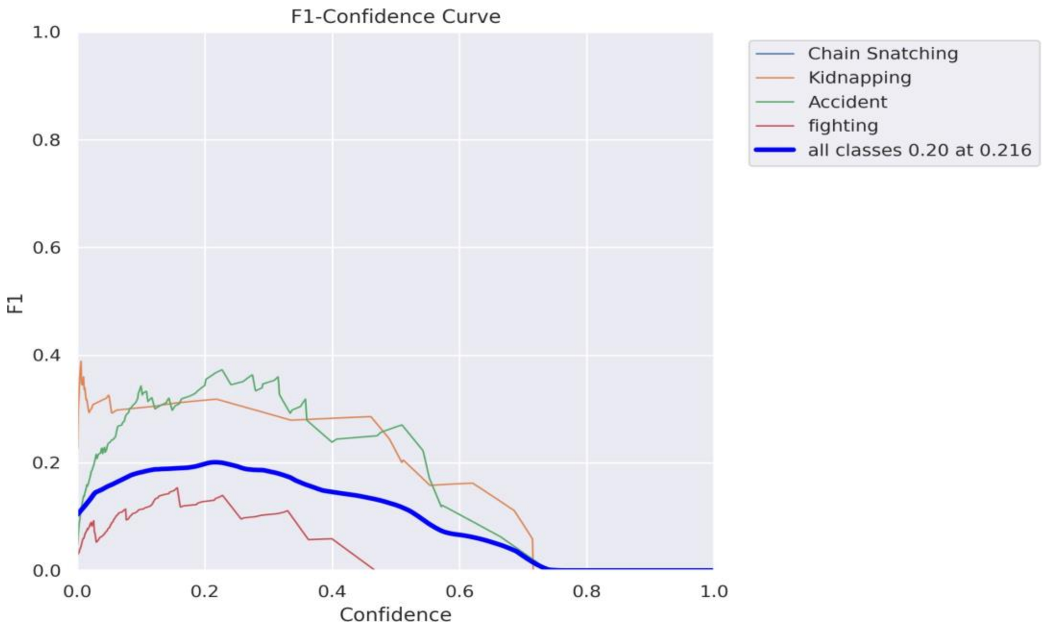

Figure 9 :F1-Confidence Curve

In Figure 9 The x-axis represents "Confidence" values ranging from 0.0 to 1.0, while the y-axis sh- ows the "F1" metric, which is a measure that combines precision and recall, ranging from 0.0 to 1.0.

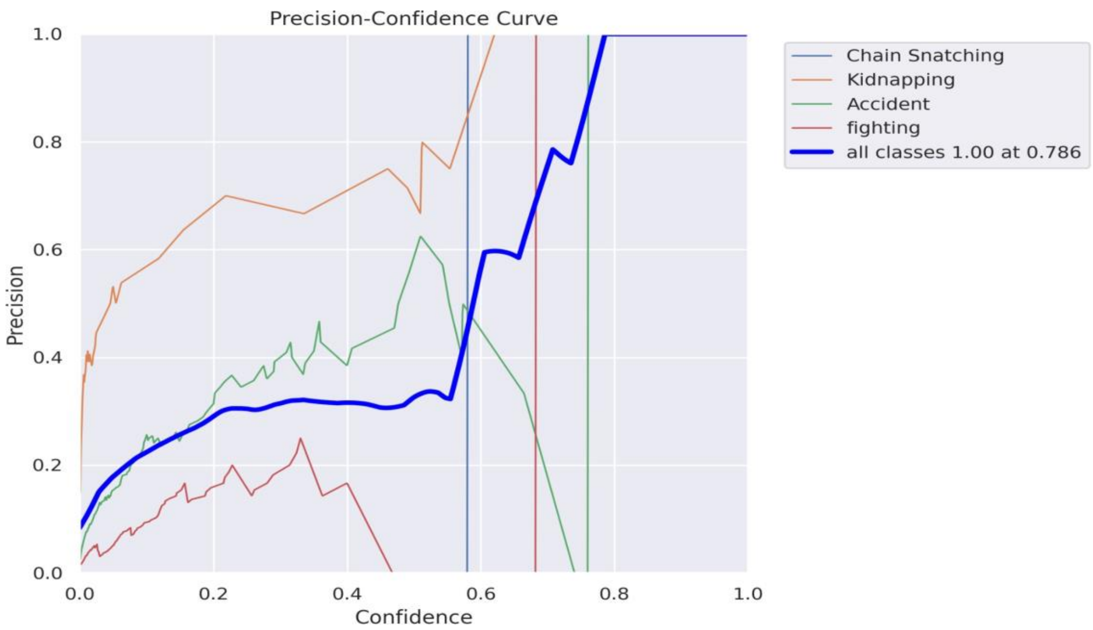

Figure 10 :Precision-Confidence Curve

In Figure 10 The x-axis represents "Confidence" values ranging from 0.0 to 1.0, while the y-axis shows "Precision" values, also ranging from 0.0 to 1.0.

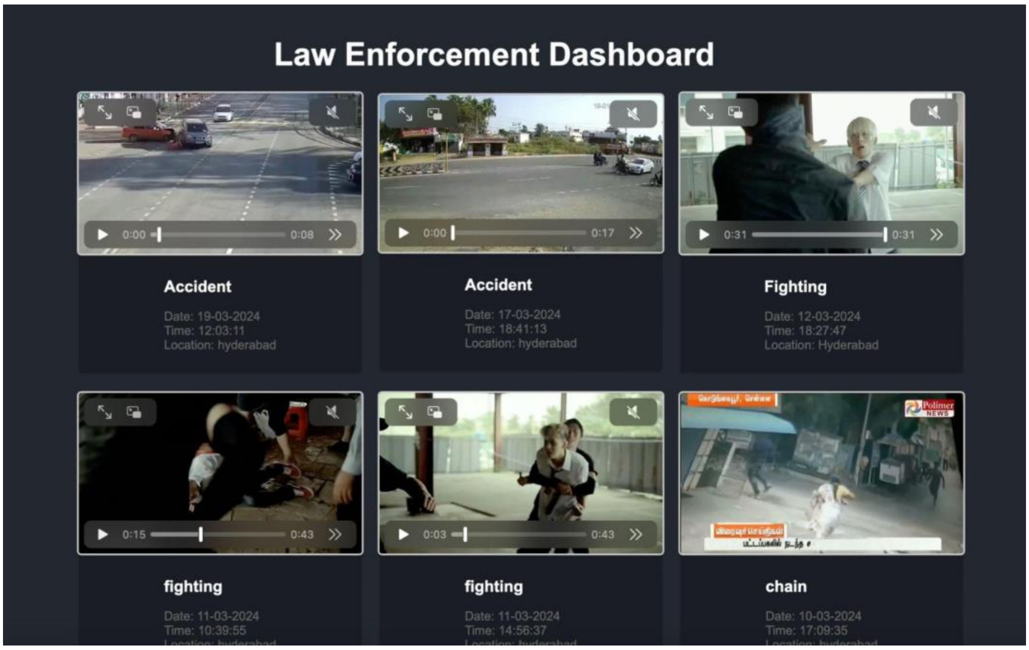

Figure 11 : Authority Dashboard

In Figure 11 Emergency Dashboard display Abnormal activities that take place on thepathway are up-loaded to Cloudinary, which the Healthcare unit accesses to rescue thesituation.

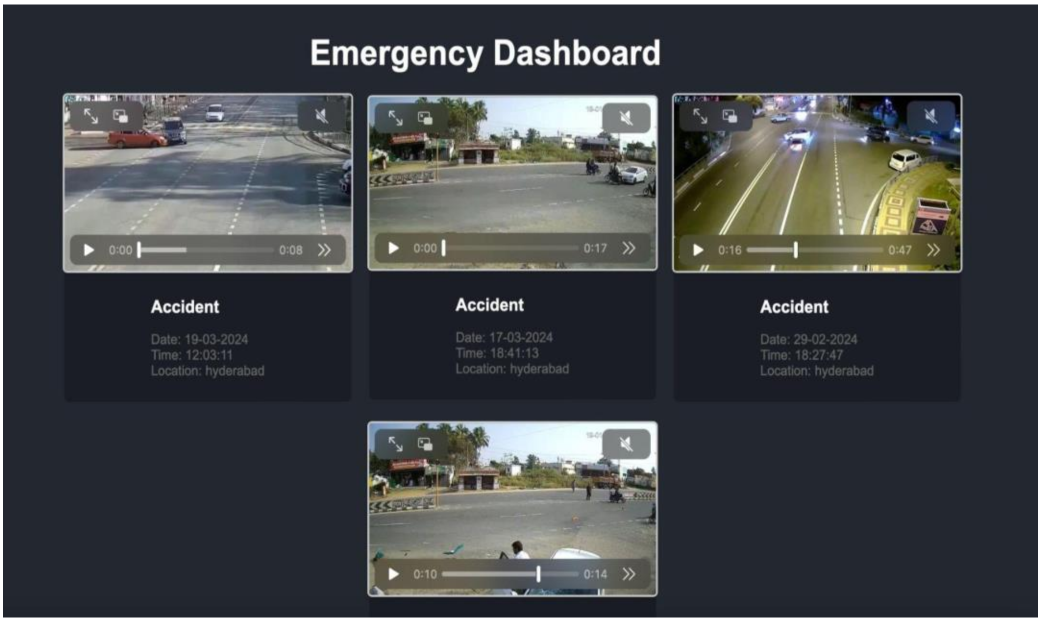

Figure 12 : Emergency Dashboard

In Figure 12 Authoritative Dashboard display Any abnormal activities that takes place on the pathway are up- loaded to cloudinary which are accessed by the concerned authority.

Fig 13 :Kidnapping Detection

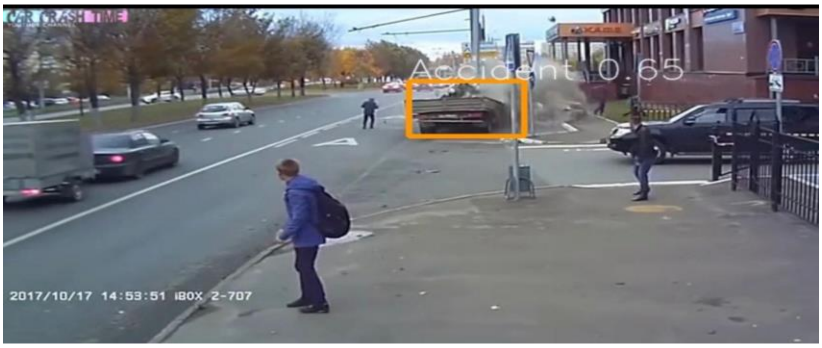

Fig 14 :Accident Detection

Fig 15: ChainSnatching Detection

Fig 16:Fighting Detection

CHAPTER 7

CONCLUSION

In conclusion, the project focuses on enhancing road safety through real-time detection of abnormalities using the YOLOv8 object detection model. By Utilizing the potent YOLO object detection model, the built video object detection and recording system has successfully exhibited the ability to detect and record events of interest such as fights, accidents, kidnapping, and chain snatching. Although the system as it is now shows encouragingoutcomes, there is a great deal of room for improvement. Future research could investigate the following expanding the scope to detect additional event types; integrating with existing security systems for a unified solution; integrating cloud storage and real-time streaming for centralized monitoring; incorporating object tracking for thorough even; optimizing for deployment on edge devices and IoT platforms; and continuously improving the model's accuracy through data collection and retraining. Through a combination of Flask and Cloudinary, the system offers user authentication, video processing, and streamingcapabilities.The YOLOv8 model demonstrates strong performance in terms of speed and accuracy when detecting traffic anomalies, achieving over 65% mean average precision on datasets like UA-DETRAC. Overall, the system shows promise in proactive identification of road hazards and contributing to overall road safety.

FUTURE ENHANCEMENTS AND DISCUSSIONS

To create a more thorough safety monitoring system, broaden the scope to identify other event types like thefts, identifying the Rule Violators, traffic violations, tracking the chain snatchers, Identifying Emergency Situations or roadblocks.Connect with current security systems to create a unified solution that enables easy data exchange and collaboration between various monitoring platforms.For centralized monitoring, include cloud storage and real-time streaming so that authorities can access recorded events and live feeds from any location.Use object tracking to do comprehensive event analysis, which can assist in figuring out trends and comprehending how issues develop over time.Scalability and flexibility in monitoring many locations are ensured by optimizing for deployment on edge devices and IoT platforms.Retraining and data gathering should be used to continuously increase the model's accuracy in order to improve the system's performance over time.