ABSTRACT

The project aims to develop an intelligent video surveillance system capable of detecting and classifying suspicious activities in real-time. By leveraging cutting-edge technologies like YOLOv8 and Vision Transformers. The system processes video inputs to identify various suspicious behaviors, such as violence, theft, vandalism, or unusual movements in restricted zones. The system integrates advanced deep learning models to analyze spatial and temporal features, ensuring accurate differentiation between benign and harmful activities, such as distinguishing between a hug and a physical fight. To enhance performance, the project employs ensemble learning techniques and adaptive thresholding to classify activities with high confidence. The system is designed for scalability and real-time processing while addressing privacy concerns through anonymized video data handling. Ultimately, this project represents a significant step forward in developing AI-powered solutions for public safety and security monitoring.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The project focuses on developing an advanced AI-driven surveillance system designed to identify and classify suspicious human activities in real-time video feeds. Traditional video surveillance methods often rely heavily on human operators, leading to inefficiencies, inaccuracies, and delays in identifying critical events. The need for an automated system arises from the increasing demand for efficient monitoring in public and private spaces, ensuring timely detection of potential threats and preventing harmful incidents.

This project leverages deep learning techniques, particularly YOLOv8, for object detection and classification. By training the model on a dataset enriched with ten predefined classes of suspicious activities, such as violence, theft, vandalism, and restricted zone breaches, the system achieves high precision in recognizing diverse scenarios. Additionally, Optical Flow is utilized to track and analyze object motion across frames, enabling the differentiation of subtle behaviors like a hug versus a physical altercation.

The system integrates spatial and temporal feature analysis, ensuring accurate identification of activities in complex environments. Advanced techniques such as Vision Transformers enhance the model's ability to process and classify dynamic video data with improved context understanding. The incorporation of adaptive thresholding and ensemble learning further optimizes classification accuracy, reducing false positives and negatives.

The project also addresses the challenges of handling various video formats, supporting uploads in mp4, avi, and other popular formats. This ensures accessibility for diverse applications, including law enforcement, workplace monitoring, and public safety. The system's scalability and real-time processing capabilities are designed to meet the growing demands of modern surveillance requirements.

In addition to technical sophistication, the project emphasizes ethical considerations, ensuring anonymized data processing and compliance with privacy regulations. By blending cutting-edge AI technologies with practical application strategies, the system provides a robust and reliable solution for proactive monitoring and threat prevention. This innovative approach

represents a significant advancement in surveillance technologies, catering to the evolving needs of safety and security in contemporary society.

1.2 DESCRIPTION

The project aims to revolutionize video surveillance systems by developing an AIpowered model capable of detecting and classifying suspicious activities in real time. Surveillance is a critical aspect of public safety, but the manual monitoring of video feeds often leads to inefficiencies, errors, and delayed responses. The primary objective of this project is to automate this process, enhancing the accuracy and timeliness of detecting potentially dangerous situations.

Traditional surveillance systems are limited in their ability to identify complex human behaviors and contextual scenarios, often resulting in false alarms or missed incidents. This project leverages cutting-edge deep learning algorithms, specifically the YOLOv8 model, to overcome these challenges. By training the model on a carefully curated dataset with ten classes of suspicious activities including violence, theft, vandalism, and unusual behaviors. The system ensures a high level of precision and adaptability.

One of the unique features of the project is the integration of Optical Flow techniques to analyze object motions across video frames. This capability is crucial in distinguishing between subtle actions such as a hug versus a fight, providing contextual awareness that traditional systems lack. Moreover, the model incorporates adaptive thresholding to dynamically adjust sensitivity based on environmental conditions, further reducing false positives and negatives.

The system is designed to handle multiple video formats, such as mp4 and avi, ensuring compatibility and usability across diverse applications. Additionally, the project emphasizes scalability and real-time processing, making it suitable for a wide range of settings, from public spaces to private premises. The ethical aspects of surveillance, such as privacy and data security, are also prioritized, ensuring the model operates within regulatory frameworks and societal norms.

This innovative approach not only enhances the capabilities of existing surveillance systems but also opens new possibilities for proactive security measures. By automating the detection of suspicious activities, the project contributes to creating safer environments while

reducing the burden on human operators. The system's versatility, combined with its advanced technical features, positions it as a transformative solution in the field of intelligent surveillance.

1.3 OBJECTIVE

The primary objective of this project is to develop an intelligent surveillance system capable of automatically detecting and classifying various suspicious human activities using advanced computer vision and deep learning technologies. The system focuses on enhancing public safety by identifying activities such as violence, theft, vandalism, and other potentially harmful behaviors with high accuracy.

The project aims to provide a real-time solution for monitoring video footage to reduce manual efforts and errors typically associated with traditional surveillance systems. By incorporating advanced classification techniques, the system can differentiate between similar actions and identify suspicious activities based on environmental and contextual cues. The solution also seeks to handle multiple video formats and dynamically adapt to various scenarios while ensuring ethical compliance and protecting individuals' privacy.

This system is envisioned as a robust and scalable solution that can be implemented across diverse environments, such as public spaces, workplaces, and restricted zones, to significantly improve security and enable swift responses to potential threats.

1.4 ORGANIZATION OF PROJECT

Chapter 1 describes the development of AI-driven surveillance system to identify and classify suspicious human activities and addresses the challenges of handling various video formats.

Chapter 2 contrast and compare the current work with earlier ones and it gives attention to and puts the focus on the upcoming work that would be undertaken based on the existing work and shows current implementations that address the prior difficulties and limitations of the suspicious activity detection system.

Chapter 3 delves into the project description and problem definition by the existing system as well as the methodology of the proposed system

Chapter 4 covers the flow diagram of the project and basic theoretical information about the Each module and numerous aspects of the task

Chapter 5 delves deeper into the system investigates the system architecture and provides a realization of the concepts and ideas developed previously and it covers the project roadmap which explains the timeline of each project module

Chapter 6 discusses the findings acquired on the degree of efficacy of the suggested design strategy of the Suspicious activity tracking system

Chapter 7 highlights the key lessons learned as well as the essential future improvements that can be employed to improve the overall viability of the proposed system

CHAPTER 2

LITERATURE SURVEY

Amrit Sen et al. [1] proposed a novel system for detecting live events associated with threats such as robbery, assault, or homicide through the analysis of surrounding audio signals. This system is designed to ensure safety, especially for individuals working alone in remote areas, by detecting suspicious sounds in real time. The research utilizes Exploratory Data Analytics (EDA) techniques to process audio data from a Kaggle dataset and employs deep learning models like Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) for classification. Their model achieves an accuracy of 96.6%, demonstrating its potential for early detection and real-time alert mechanisms via smartphone notifications, including emails, SMS, and WhatsApp messages. This approach highlights the utility of sound as an effective input for threat detection but faces limitations in noisy environments or cases of ambiguous audio patterns.

Abdul Aziz A.B and Aindri Bajpai et al. [2] proposed an advanced intelligent surveillance system utilizing YOLOv8 for anomaly detection in restricted areas. The system focuses on attire-based identification to monitor unauthorized access, addressing traditional security challenges in restricted zones. The YOLOv8 model was trained on a dataset of uniform patterns, enabling precise recognition of authorized personnel. Soft computing techniques were integrated to improve adaptability under dynamic environmental conditions, such as lighting variations. This study demonstrated YOLOv8's potential for creating robust, uniform-based surveillance systems, enhancing security measures in sensitive locations.

Sudharson D et al. [3] developed a proactive headcount and suspicious activity detection system utilizing YOLOv8 for managing public safety in crowded spaces. The system addresses challenges like crowd-smashing accidents by providing real-time crowd density evaluations and issuing instant alerts to authorities. It extends beyond crowd counting by detecting abnormal activities, including weapons, fires, falls, and smoke within the crowd. This approach ensures rapid identification of potentially dangerous situations, enabling preventative measures to avoid disasters. By leveraging the capabilities of YOLOv8, the system strengthens public safety protocols and enhances crowd management in response to the growing global population and its associated challenges.

Sanskar Singh et al. [4] proposed a violence detection system using Video Vision Transformers (ViViT) to enhance law enforcement and public safety through automated surveillance. This approach addresses the limitations of traditional CCTV monitoring, where delayed human intervention often exacerbates the impact of violent events. The system employs ViViT, a deep learning-based transformer model, to accurately identify violent behaviors such as fights and hostile movements in video sequences. Data augmentation techniques were utilized to address the challenges posed by smaller datasets, reducing biases during training. When evaluated on benchmark datasets, the model demonstrated superior performance compared to state-of-the-art methods. This system ensures timely detection of violent events and efficient communication of alerts to local authorities, significantly improving response times and mitigating risks to people and property.

Kapil Deshpande et al. [5] introduced a weakly supervised framework for detecting anomalies in surveillance videos, addressing the limitations of traditional supervised methods requiring frame-level annotations. This approach utilizes only video-level labels, enabling efficient anomaly score predictions for individual frames. To overcome the challenges of misclassifying normal and abnormal instances during training, the model employs Video Swin transformer-based features combined with an attention mechanism using dilated convolutions and self-attention. This integration captures both short- and long-range temporal dependencies, resulting in better feature extraction. Evaluated on the ShanghaiTech Campus dataset, the framework achieves competitive performance against state-of-the-art methods. The model's implementation and codebase are available for public use, encouraging further research and practical adoption in anomaly detection systems.

Muhammad Shoaib et al. [6] proposed robust and efficient violence detection systems tailored for both surveillance and non-surveillance scenarios. Their research emphasizes the significance of automated systems in ensuring public safety by promptly identifying violent incidents in vast video databases. Two keyframe-based models, DeepKeyFrm and AreaDiffKey, were introduced for optimal keyframe selection, reducing computational complexity while enhancing detection accuracy. Additionally, classification models like EvoKeyNet, which uses evolutionary algorithms for feature selection, and KFCRNet, an ensemble of LSTM, Bi-LSTM, and GRU with a voting mechanism, were developed the contribution of these approaches towards advanced violence detection systems that balance accuracy and efficiency for dynamic environments.

Manisha Mudgal et al. [7] proposed a system for suspicious action detection in intelligent surveillance systems using action attribute modeling. This research highlights the challenges faced by human operators in continuously monitoring surveillance videos, especially in sensitive locations like airports, railway stations, and banks. The study emphasizes the identification of violence-related activities such as hitting, slapping, and punching. The methodology leverages a Gaussian Mixture Model (GMM) with a Universal Attribute Model (UAM) to represent violence-related activities. A Super Action Vector (SAV) is computed using UAM, and factor analysis is employed to reduce the dimensionality, isolating relevant attributes for each action. The system demonstrates its effectiveness in recognizing abnormal activities in real-time using datasets such as UCF101 and Kaggle.

Kunal Kamble et al. [8] developed a smart surveillance system aimed at detecting anomalous behavior in video footage, addressing the challenge of manually monitoring large volumes of surveillance data. Their proposed framework integrates a deep neural network, consisting of three core modules: (a) object detection, (b) object discriminator and tracking, and (c) anomalous activity detection using recurrent neural networks (RNN). This system is designed to function in various environments such as homes, roads, offices, schools, and shops, offering security services like motion detection, fall detection, and anomaly detection. The system sends instant notifications to users and relevant authorities, along with an attachment of the detected anomalous activity. This framework emphasizes efficient and autonomous anomaly detection, minimizing human intervention in security systems.

Jungpil Shin et al. [9] propose a multi-stage deep learning model for weakly supervised video anomaly detection (WS-VAD), addressing the challenges of feature extraction in videobased anomaly detection systems. The model utilizes two distinct streams for feature extraction: the first stream uses a ViT-based CLIP module to select top-k features, while the second stream incorporates a CNN-based I3D module combined with Temporal Contextual Aggregation (TCA) to capture temporal information. These features are then processed through an Uncertainty-regulated Dual Memory Units (UR-DMU) model, which integrates Graph Convolutional Networks (GCN) and Global and Local Multi-Head Self Attention (GL-MHSA) modules for capturing video associations. The proposed method achieves significant performance improvements on datasets such as ShanghaiTech, XD-Violence, and UCF-Crime, validating its superiority over state-of-the-art approaches. The model offers an effective solution to anomaly detection, enhancing feature extraction and enabling robust performance in real-world surveillance applications.

Shubham Kale et al. [10] propose a smart surveillance system tailored for Indian cities, leveraging artificial intelligence and machine learning to identify and analyze individual attributes in real-time. This system is designed to recognize various personal features, such as upper body color, clothing, accessories, and headgear, using cameras placed throughout urban areas. The project focuses on developing a comprehensive monitoring solution for urban environments, allowing real-time behavior analysis. The team provides open access to their code through a GitHub repository, ensuring that their work remains updated with the latest advancements in person attribute recognition. This research contributes to improving urban security and management by enabling enhanced surveillance capabilities.

Asim Niaz et al. [11] propose an innovative unsupervised video anomaly detection model that addresses the challenges of defining abnormal events in surveillance footage. Their architecture utilizes a 3D convolutional autoencoder that learns spatiotemporal features to detect anomalies. The model employs skip connections between encoder and decoder blocks to transfer feature representations across different scales, enhancing the accuracy of the anomaly detection. Spatial attention modules are incorporated to highlight key regions in the input frames, improving the system's ability to focus on the most relevant areas. The model's effectiveness is demonstrated through its performance on four benchmark datasets: UCSD Pedestrian 1, UCSD Pedestrian 2, CUHK Avenue, and ShanghaiTech, achieving AUC scores of 94.6%, 96.7%, 84.7%, and 74.8%, respectively. This approach pushes the boundaries of real-time anomaly detection, making it a promising solution for surveillance systems in diverse environments.

Jonghwan Hong et al. [12] introduced SYRFA (SYnthetic-to-Real via Feature Alignment), a novel framework for Video Anomaly Detection (VAD) that addresses the challenge of domain shift between synthetic and real-world datasets. Many existing methods rely on generation models to synthesize abnormal events, but these models are often computationally expensive. The SYRFA framework, on the other hand, avoids such models by focusing on feature alignment between synthetic and real domains, thereby reducing computational burden. It operates in two learning phases: learning synthetic knowledge and adapting to the real-world domain. The key contribution is consistency learning, which aligns feature representations to bridge the domain gap. Additionally, the Residual Additional Parameters (RAP) method enhances local pattern learning, allowing for more transferable features with minimal computational cost. The framework has demonstrated superior performance on benchmark VAD datasets, including outperforming other methods by 0.8% on

the ShanghaiTech dataset. This approach stands out for its ability to handle domain shifts without heavy computational demands, offering a promising solution for more efficient and effective anomaly detection in surveillance systems.

Indhumathi J et al. [13] proposed a Real-Time Human Suspicious Activity Recognition System using deep learning techniques to identify and differentiate between suspicious and normal human behavior. The system employs a 2D-CNN, VGG16, and ResNet50 for activity recognition, focusing on criminal, suspicious, and normal behavior categories. The model is trained using real-time video frames of human activities, with and without transfer learning. The study found that ResNet50 achieved the highest accuracy of 99.03% without transfer learning, while with transfer learning, it achieved an accuracy of 99.18%, outperforming VGG16, which had an accuracy of 98.36%. The proposed system demonstrated high performance in classifying suspicious activities in both controlled datasets and real-time videos, with the 2D-CNN model outperforming VGG16 on real-time videos from Kaggle. This work showcases the effectiveness of leveraging pre-trained models and transfer learning for human activity recognition, providing a reliable solution for real-time surveillance and anomaly detection in various environments.

Mir Ali Rezazadeh Baee et al. [14] proposed a framework for Anomaly Detection in Key-Management Activities Using Metadata within Enterprise Key-Management Systems (EKMS). The research focuses on detecting anomalous activity by analyzing EKMS metadata, a largely unexplored area compared to system-level log analysis. The framework integrates automated outlier rejection, heuristic generation, anomaly detection, and system notifications, which are key to identifying potentially malicious behaviors that other security systems might miss. The authors employed a Long Short-Term Memory (LSTM) auto-encoder deep learning model to process the generated datasets and detect patterns of normal and abnormal behavior. The proposed approach was tested in a simulated enterprise environment using QuintessenceLabs EKMS, with results demonstrating its efficacy in accurately identifying anomalous activities. The framework is vendor-neutral, offering a scalable solution for enhancing anomaly detection and can be integrated with other security systems for improved capabilities. This work provides a generalized framework applicable to various enterprise environments, promising significant advancements in network security.

Kian Yu Gan et al. [15] introduced a Contrastive-Regularized U-Net architecture for Video Anomaly Detection, focusing on two significant challenges: capturing both local and

global temporal dependencies and mitigating overfitting due to limited training data. Their work improves upon previous models that captured either local or global dependencies but struggled to combine both. The U-Net-like structure proposed in this study effectively models these dependencies, with the encoder learning global temporal information hierarchically while the decoder propagates this knowledge back to the segment level for classification. To address overfitting, the authors propose a weakly supervised contrastive regularization technique. This feature-based approach enhances the model's generalizability by promoting inter-class separability and intra-class compactness. Extensive experiments on the UCF-Crime dataset showed that the proposed method outperforms existing state-of-the-art video anomaly detection models, marking a significant advancement in anomaly detection performance for surveillance systems.

CHAPTER 3

PROBLEM DEFINITION AND METHODOLIGIES

3.1 PROBLEM DEFINITION

Monitoring and ensuring public safety in high-risk areas has become increasingly challenging due to the limitations of traditional surveillance systems. The reliance on manual monitoring of video footage is not only time-consuming but also prone to errors due to human fatigue and oversight. As a result, identifying and responding to suspicious activities such as violence, theft, or vandalism in real time becomes inefficient, leading to delayed interventions and compromised security.

Existing surveillance systems often fail to differentiate between contextually similar activities, such as friendly physical interactions and hostile altercations, due to the absence of advanced classification capabilities. Furthermore, the lack of automated processes in traditional systems increases dependency on constant human supervision, which is resource-intensive and unsustainable in large-scale surveillance scenarios.

The need arises for an intelligent system capable of analyzing video footage in real time, detecting anomalies, and classifying them into predefined categories of suspicious activities. Such a system should also adapt dynamically to diverse environmental contexts and reduce false alarms while ensuring privacy and ethical standards. Addressing these challenges requires integrating advanced machine learning and computer vision techniques to create a robust, scalable, and accurate solution.

3.2 EXISTING SYSTEM

Existing systems for real-time event detection, particularly those aimed at enhancing public safety, have primarily relied on traditional hardware-based surveillance and detection mechanisms, such as CCTV cameras and sound sensors. These systems are designed to monitor environments and identify potential threats based on predefined criteria like motion detection or decibel levels for unusual sounds. While these methods have been effective in some scenarios, they face significant limitations in terms of accuracy and real-time responsiveness.

The primary limitation of hardware-based systems is their dependency on manual monitoring or basic threshold-based detection, which often leads to delayed responses or false alarms. For instance, sound-based systems may struggle to differentiate between noises like gunshots and fireworks, leading to misinterpretations. Similarly, traditional CCTV systems require constant human supervision, making them prone to oversight and inefficiency, especially in high-risk or crowded areas.

In recent years, some advancements have introduced artificial intelligence (AI) and machine learning (ML) techniques into event detection systems. These include image recognition for detecting suspicious activities and audio signal processing for identifying threatening sounds. However, the majority of these systems focus on isolated data types (either visual or audio) rather than integrating multiple sources for holistic threat detection. Moreover, existing systems often fail to address contextual analysis, which is crucial for reducing false positives and accurately classifying events in dynamic real-world environments.

One example of an existing system is the use of Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) for audio classification. These models achieve significant accuracy but are limited in their application to real-time scenarios due to computational delays. Similarly, Natural Language Processing (NLP) has been used in some systems to analyze textual descriptions of events; however, this approach is not feasible for real-time detection in noisy environments.

While these systems represent a step forward in leveraging AI for safety, they are not without challenges. The lack of integration between visual and audio data, reliance on static datasets, and limited adaptability to evolving threats make them insufficient for comprehensive safety applications. These shortcomings highlight the need for a robust, integrated solution that combines multiple data sources, utilizes advanced deep learning models, and ensures real-time processing with minimal delays.

Hence, the existing systems, while innovative, are often inadequate for addressing the complexity and urgency of real-world safety scenarios. They pave the way for more advanced, integrated approaches, such as the proposed system, which leverages cutting-edge AI and deep learning techniques to enhance both the accuracy and efficiency of threat detection mechanisms.

3.3 PROPOSED SYSTEM

The proposed system aims to overcome the limitations of existing event detection mechanisms by leveraging the state-of-the-art YOLOv8 (You Only Look Once, version 8) deep learning model for real-time threat detection. YOLOv8 is a cutting-edge object detection and image classification framework known for its speed, accuracy, and ability to operate efficiently in real-world conditions.

The proposed system integrates YOLOv8 with advanced feature extraction techniques to analyze video feeds and detect suspicious activities in real time. YOLOv8's capability to process high-resolution images and detect multiple objects in a single frame significantly reduces delays in threat identification, addressing the computational latency observed in existing systems.

Advantages of YOLOv8 in the Proposed System

- Enhanced Detection Accuracy:
  YOLOv8 is equipped with improved neural architecture and anchor-free object detection, which enhances its ability to detect complex and overlapping objects. This ensures accurate identification of threats such as violent activities, unauthorized access, or dangerous objects, thereby reducing false positives and false negatives.
- Real-Time Processing:
  Unlike existing systems that may experience delays, YOLOv8 processes video streams at high speeds without compromising accuracy. This ensures timely detection and alert generation, making the system highly effective in dynamic environments.
- Contextual Awareness:
  The model utilizes spatial and contextual cues within the video frames to differentiate between benign and suspicious activities. For instance, it can discern a fight from normal interactions based on motion patterns and object trajectories.
- Scalability and Adaptability:
  YOLOv8's flexible framework allows it to adapt to various environments, whether it is monitoring public spaces, offices, or remote areas. The system can be trained on diverse datasets to improve detection accuracy for specific use cases.

3.4 ALGORITHM OF THE PROPOSED SYSTEM

3.4.1 YOLOv8 Algorithm

The YOLOv8 (You Only Look Once, Version 8) model is an advanced object detection framework that performs real-time detection and classification tasks. It is designed for high-speed processing and accuracy, suitable for detecting suspicious activities in video surveillance. YOLOv8 processes images or video frames in a single pass through the network, enabling rapid inference while maintaining precision.

Input:

Video footage or image frames from CCTV cameras or other surveillance systems.

Output:

Classified detections of suspicious or abnormal activities, annotated with bounding boxes and confidence scores for each detected activity.

Step 1: Dataset Preparation

The dataset is prepared using Roboflow, where images and video frames containing labeled abnormal activities are processed. The data is then split into training, validation, and test sets.

Step 2: Model Initialization

The pre-trained YOLOv8 model (yolov8m.pt) is selected as the starting point for training and fine-tuning to detect suspicious activities.

Step 3: Training Configuration

The model is configured for training with a batch size suitable for the hardware, an image resolution of 640x640, and 10 epochs.

Step 4: Model Training

The YOLOv8 model is trained on the prepared dataset to learn features that differentiate normal and abnormal activities. Training generates weight files like best.pt and last.pt.

Step 5: Validation

The model is validated using the validation set, producing evaluation metrics like precision, recall, and mAP (mean Average Precision). Visualization tools display confusion matrices and validation results.

Step 6: Testing the Model

The trained model is tested on unseen data from the test set or new surveillance footage to verify its performance in detecting anomalies.

Step 7: Prediction on Real Data

The best-trained weights (best.pt) are utilized for predictions on real-world data, including images and videos. Results are displayed with bounding boxes and labels for suspicious activities.

Step 8: Visualization of Results

Output visualizations include:

- Confusion matrices showcasing detection performance.
- Annotated images and video frames highlighting detected abnormalities.
- Summary plots of training progress.

Step 9: Model Optimization

If the initial results are not satisfactory, hyperparameters such as learning rate and batch size are tuned. The model is re-trained or fine-tuned as necessary.

Step 10: Deployment

The trained YOLOv8 model is integrated into the surveillance system for live video monitoring and automatic detection of suspicious activities.

3.4.2 Vision Transformer model

The Vision Transformer (ViT) model is a state-of-the-art deep learning framework designed for image classification and feature extraction tasks. By leveraging the self-attention mechanism from transformers, ViT efficiently processes input images to detect anomalies and classify suspicious activities.

Input:

Images or video frames extracted from surveillance systems.

Output:

Predictions with labels and confidence scores indicating suspicious or normal activities.

Step 1: Dataset Preparation

Images and video frames containing labeled suspicious activities are collected and organized into training, validation, and test sets.

Step 2: Model Initialization

The Vision Transformer (ViT-B/16) model pre-trained on ImageNet is selected as the starting point and Pretrained weights are loaded, and the model is moved to the appropriate hardware (CPU/GPU).

Step 3: Feature Extraction and Fine-tuning

The base parameters of the pre-trained ViT model are frozen to retain learned features and the classifier head is replaced with a fully connected layer tailored to the number of activity classes (e.g., suspicious or normal).

Step 4: Training Configuration

The training process is configured with a suitable batch size, learning rate, and number of epochs and a cross-entropy loss function is employed for classification tasks, optimized using the Adam optimizer.

Step 5: Model Training

The ViT model is trained on the prepared dataset then during training, the model learns to identify patterns and distinguish between normal and suspicious activities

Step 6: Validation

The model is validated using the validation set to evaluate its generalization capability and results are visualized using confusion matrices and performance plots.

Step 7: Testing the Model

The trained model is tested on unseen data to assess its ability to detect suspicious activities in real-world scenarios.

Step 8: Real-World Predictions

The best-performing model weights are used for predictions on new images or video frames and Output includes class labels (e.g., suspicious or normal) and confidence scores.

Step 9: Visualization of Results

Annotated images or video frames highlighting detected activities.

Step 10: Model Optimization

If performance is unsatisfactory, hyperparameters such as learning rate and number of layers in the classifier head are adjusted and further fine-tuning is performed to enhance model accuracy and robustness.

Step 11: Deployment

The fine-tuned ViT model is integrated into the surveillance system for real-time monitoring and the system continuously processes incoming video frames, classifying activities as normal or suspicious with high accuracy.

CHAPTER 4

DESIGN PROCESS

4.1 FLOW DIAGRAM

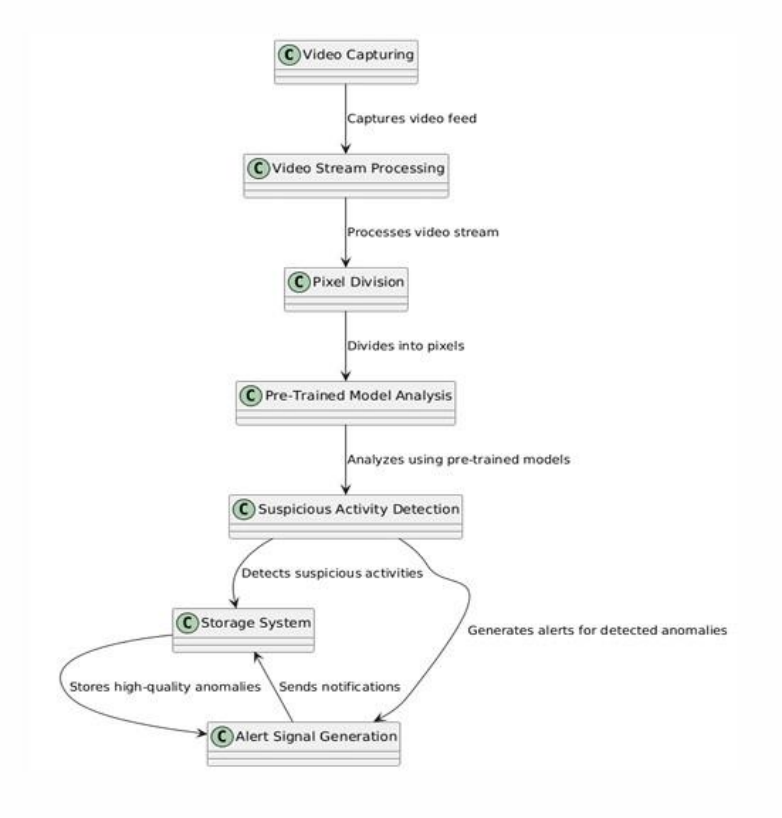

figure 4.1 Flow diagram

The flow diagram illustrates the operational workflow of the system, breaking it into key functional blocks. The Video Capturing module acquires footage from surveillance environments where suspicious activities are monitored. The Video Stream Processing block transforms raw video frames into pixel data, compressing them into standardized dimensions, such as 224x224. Pre-trained Models, including YOLOv8 and the Vision Transformer, are utilized to detect abnormal events effectively. Upon detection, the system engages the Storage Module to archive the detected clips securely. Simultaneously, the Alert Signal Generation module issues real-time alerts to designated emergency locations, facilitating live-stream surveillance for prompt response.

4.2 SYSTEM REQUIREMENTS

4.2.1 HARDWARE REQUIREMENTS

The specifications of the hardware components used in the proposed system are shown in table

HARDWARE	SPECIFICATION
Processor	Core i5 2.4 Ghz
Hard disk	Min. 250 GB HDD
RAM	Min. 8 GB RAM

4.2.2 SOFTWARE REQUIREMENTS

The specifications of the hardware components used in the proposed system are shown in table

SOFTWARE	SPECIFICATION
Operating System	Windows 10 or higher
Language	Python
Packages	Ultralytics, roboflow, Ipython display,
IDE	Google Colab, Visual studio code

4.3 SOFTWARE DESCRIPTION:

4.3.1 GENERAL:

This section provides an overview of the tools, frameworks, and programming languages utilized in the development of the project. The proposed system is implemented using Python, leveraging its powerful libraries and frameworks for deep learning, computer vision, and real-time object detection.

4.3.3 THE PYTHON ECOSYSTEM

Python is a high-level, interpreted programming language known for its simplicity, versatility, and extensive ecosystem of libraries and frameworks. Developed by Guido van Rossum and first released in 1991, Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Its platform-independent nature allows for seamless deployment across diverse environments.

Key Features of Python for the Project:

- Versatility: Python can handle tasks ranging from data preprocessing to real-time object detection.
- Extensive Libraries: Libraries such as TensorFlow, PyTorch, OpenCV, and NumPy are utilized to implement and optimize deep learning models.
- Community Support: Python's vast community ensures constant updates, resources, and troubleshooting assistance.

4.3.4 OPENCV: AN OVERVIEW

OpenCV (OpenSource Computer Vision Library) is an open-source computer vision library that provides tools for image and video processing. It supports a wide range of applications, including object detection, image enhancement, and facial recognition.

Advantages of OpenCV in the Project:

- Real-time video processing.
- Integration with Python for seamless implementation.
- Support for various video and image formats.

4.3.5 TENSORFLOW AND YOLOv8 FRAMEWORK

TensorFlow is an open-source machine learning framework developed by Google. It is widely used for deep learning model development and deployment. TensorFlow Hub enables the implementation of neural networks, such as the Vision Transformer and YOLOv8, used in the project.

YOLOv8 (You Only Look Once, Version 8) is the latest iteration of the YOLO series, known for its real-time object detection capabilities. It is implemented in this project for detecting suspicious human activities in surveillance videos.

Features of YOLOv8:

- Speed and Accuracy: Optimized for real-time applications.
- Customizability: Supports fine-tuning for specific datasets.
- Ease of Use: Offers pre-trained weights (yolov8m.pt) and supports transfer learning.

4.3.6 DEVELOPMENT TOOLS AND ENVIRONMENTS

- Jupyter Notebook: An interactive environment for writing and debugging Python code.
- Roboflow: Used for dataset preparation, including labeling, augmenting, and splitting data.
- Google Colab: A cloud-based platform for training deep learning models with GPU support.

4.3.7 SYSTEM IMPLEMENTATION

The system is designed to detect and classify suspicious activities in video footage using the following components:

- Pre-trained YOLOv8 weights: Enhance detection accuracy through transfer learning.
- Visualization Tools: OpenCV for rendering annotated video frames and confusion matrices for performance evaluation.
- Hyperparameter Optimization: Adjustments to learning rate, batch size, and training epochs for improved model performance.

4.4 MODULE EXPLANATION

The various modules present in the proposed system are:

1.Dataset Preparation Module

1.1 Dataset Collection

1.2 Dataset Annotation and Augmentation

1.3 Dataset Preprocessing

2.Activity Detection Module

2.1 YOLOv8 Implementation for Object Detection

2.2 Vision Transformer for Activity Classification

3.Output Analysis and Visualization Module

3.1 Prediction Results Visualization

3.2 Metrics Computation and Visualization

4.4.1 Dataset Preparation Module

This module handles the preparation of the dataset, which is a crucial step in ensuring the efficiency of the detection and classification models.

4.4.1.1 Dataset Collection

Input: A collection of video and image datasets related to suspicious human activities such as vandalism, theft, and fighting.

Methodology: Datasets are sourced from public repositories, custom CCTV footage, or simulation videos. Relevant activities are categorized and labeled based on predefined classifications.

Output: A curated dataset categorized into labels such as fighting, theft and stabbing stored in a structured format.

4.4.1.2 Dataset Annotation and Augmentation

Input: Collected video frames or image data.

Methodology: Annotation tools like Roboflow are used to label frames with bounding boxes and activity categories. Data augmentation techniques, such as rotation, flipping, and brightness adjustment, are applied to improve model generalization.

Output: Augmented and annotated dataset ready for model training.

4.4.1.3 Dataset Preprocessing

Input: Annotated dataset with raw video and image files.

Methodology: Preprocessing steps include resizing images, normalizing pixel values, and converting video frames into individual images for training. Metadata is generated to align with model input requirements.

Output: Pre-processed dataset compatible with YOLOv8 and Vision Transformer architectures.

4.4.2 Suspicious Activity Detection Module

This module detects and classifies suspicious activities using advanced AI techniques.

4.4.2.1 YOLOv8 Implementation for Object Detection

Input: Pre-processed image data from the dataset.

Methodology: YOLOv8 is trained on the dataset to detect objects like humans, weapons, or other suspicious items. The model generates bounding boxes for objects in video frames or images.

Output: Detected objects with bounding boxes and confidence scores.

4.4.2.1 Vision Transformer for Weapon classification

4.4.3 Output Analysis and Visualization Module

This module focuses on interpreting and visualizing the results obtained from the detection and classification stages.

4.4.3.1 Prediction Results Visualization

Input: Classified activity labels and bounding boxes from the Activity Detection Module.

Methodology: Results are overlaid on the input video frames or images to display detected objects and activity classifications. Visualization tools like Matplotlib or custom dashboards are utilized.

Output: Annotated video or image frames showcasing detected suspicious activities.

4.4.3.2 Metrics Computation and Visualization

Input: Model predictions and ground truth data from the validation set.

Methodology: Metrics such as precision, recall, F1-score, and confusion matrix are computed to evaluate model performance. These metrics are visualized for detailed analysis.

Output: Graphs and charts representing model accuracy and performance metrics.

CHAPTER 5

IMPLEMENTATION

5.1 ARCHITECTURE DIAGRAM

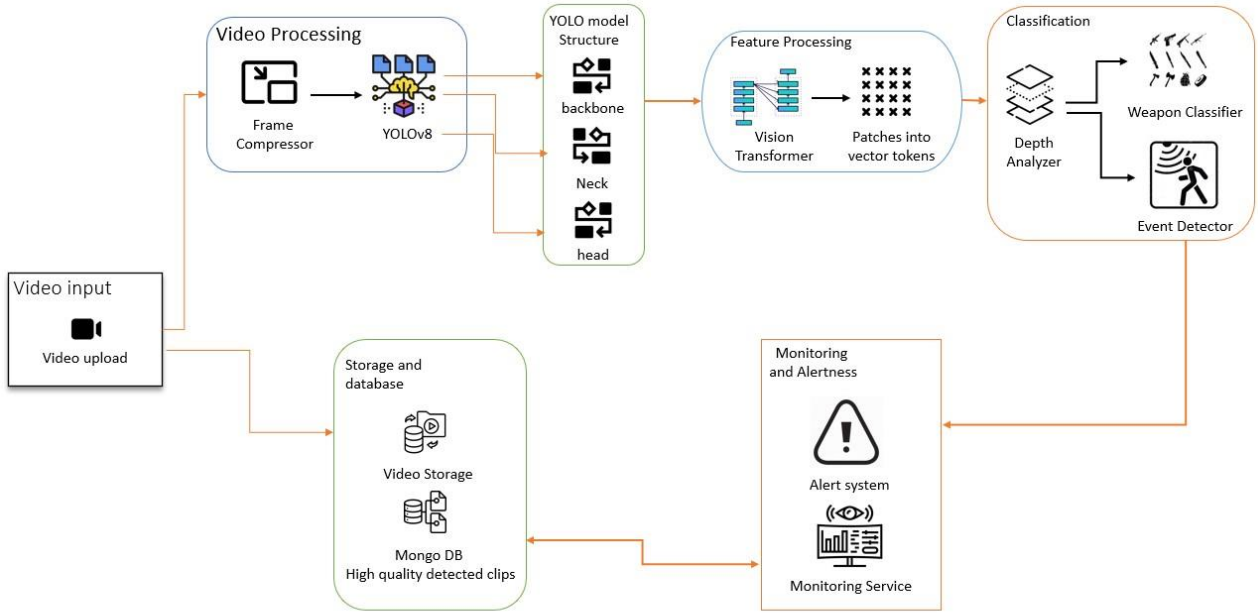

Figure 5.1 Architecture diagram

The architecture diagram of the Suspicious Activity Tracking System is shown in Figure 5.1. The system begins with the dataset preparation process, where video and image data related to suspicious activities are collected, annotated, and pre-processed. These datasets are stored in a structured format for model training and evaluation.

In the activity detection module, the YOLOv8 model takes pre-processed input data to perform object detection. It identifies objects of interest, such as humans or weapons, and generates bounding boxes with confidence scores. The features extracted from these detections are then passed to the Vision Transformer model for activity classification, where the temporal and spatial patterns are analyzed to classify actions such as fighting, walking, or theft.

The output analysis and visualization module handles the results generated by the activity detection module. The detected bounding boxes and activity classifications are overlaid on video frames or images for visualization. Evaluation metrics such as precision, recall, and

F1-score are computed using validation data to assess model performance. These metrics are represented as charts and graphs for detailed analysis.

5.2 YOLOv8 MODEL ALGORITHM

Step 1: Prepare the dataset using Roboflow, ensuring proper annotation and data splitting for training, validation, and testing.

Step 2: Initialize the YOLOv8 model with pre-trained weights (yolov8m.pt) and configure hyperparameters such as batch size and epochs.

Step 3: Train the model on the annotated dataset, learning spatial and contextual features for object and activity detection.

Step 4: Evaluate the model on the validation dataset to calculate precision, recall, F1score, and mAP metrics.

Step 5: Fine-tune the model using custom dataset weights and save the optimized weights as best.pt**.**

Step 6: Perform inference by detecting activities in real-time from video or image inputs and output predictions.

Step 7: Apply Non-Maximum Suppression (NMS) to remove redundant bounding boxes and retain accurate detections.

Step 8: Visualize results by displaying bounding boxes, activity labels, and confidence scores on video/image frames.

Step 9: Deploy the trained YOLOv8 model with live CCTV feeds and integrate alert mechanisms for detected suspicious activities.

5.3 Vision Transformer ALGORITHM

Step 1: Prepare the dataset by ensuring proper annotation and data splitting for training, validation, and testing, focusing on weapon classification.

Step 2: Initialize the Vision Transformer (ViT) model with pre-trained weights and configure hyperparameters such as batch size, learning rate, and number of epochs.

Step 3: Preprocess input images into fixed-size patches (e.g., 16x16) and embed them into linear vectors for input into the transformer.

Step 4: Train the model on the annotated dataset, leveraging the self-attention mechanism to learn spatial and contextual relationships for effective feature extraction.

Step 5: Evaluate the model on the validation dataset by calculating classification metrics such as accuracy, precision, recall, F1-score, and confusion matrices.

Step 6: Fine-tune the model using custom dataset weights to optimize its performance for the specific task of weapon classification.

Step 7: Perform inference by classifying weapons in real-time from video or image inputs, providing class predictions (e.g., knife, hammer, fist) with confidence scores.

Step 8: Visualize results by displaying class labels and confidence scores on detected weapon images or frames.

Step 9: Deploy the trained Vision Transformer model in surveillance systems to classify weapons and integrate alert mechanisms for real-time detection of suspicious activities.

5.4 PROJECT ROADMAP

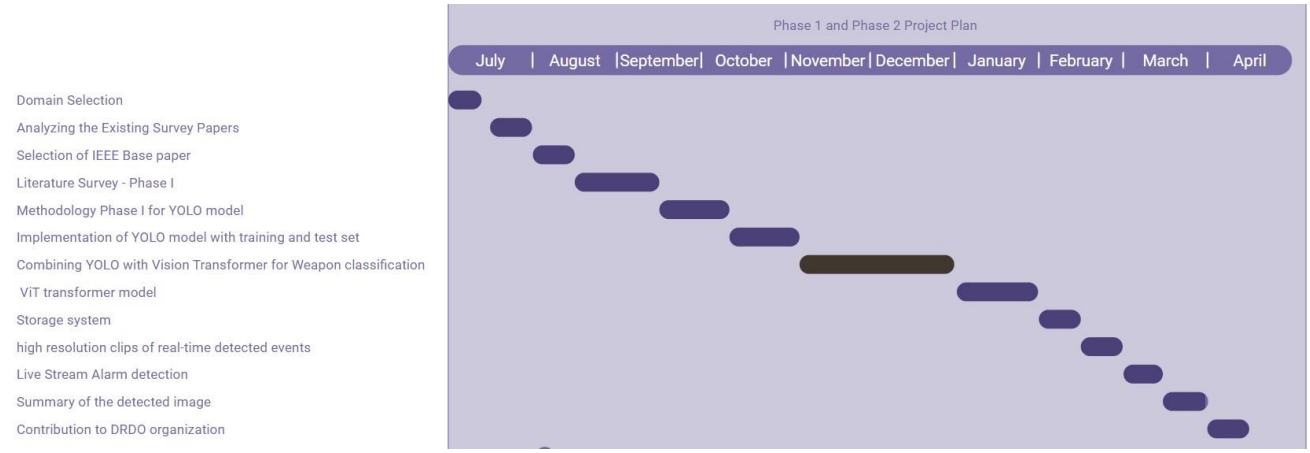

Figure 5.4 Project roadmap

The project roadmap outlines the progression through two distinct phases. Phase I spans from July to December and focuses on foundational activities, including domain selection and conducting a comprehensive literature survey to identify existing methodologies relevant to the project. This phase also involves implementing two pre-trained models, YOLOv8 and Vision Transformer, to establish individual detection capabilities. Phase II emphasizes the integration of these models into a unified workflow architecture, aiming to enhance the system's performance in detecting suspicious activities with greater accuracy and reliability.

CHAPTER 6

RESULT AND ANALYSIS

The proposed Suspicious Activity Tracking System combining YOLOv8 and Vision Transformer demonstrated robust performance in detecting and classifying suspicious activities and weapons with high precision, recall, and efficiency. YOLOv8 excelled in identifying activities like theft, fighting, and vandalism under varying conditions, while the Vision Transformer accurately classified weapons such as knives and hammers using its advanced self-attention mechanism. Minimal misclassifications and strong performance metrics, including precise confusion matrices, highlighted the system's reliability. Visualizations with bounding boxes, activity labels, and classification confidence scores enhanced monitoring capabilities. Both models showed consistent learning progress and adaptability to diverse scenarios, including low-light and crowded environments, offering a comprehensive, scalable solution for real-time surveillance and actionable insights.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

The first phase of this project has successfully demonstrated the capability of detecting and classifying suspicious activities in video footage with high accuracy and efficiency. By leveraging the strengths of YOLOv8 for real-time object detection and the Vision Transformer for detailed feature extraction, the system showed promising results in identifying activities such as theft, vandalism, and fighting, as well as accurately classifying weapons like knives and hammers.

The integration of these advanced AI techniques allowed the system to process video data efficiently while maintaining a high detection rate and minimizing false positives. YOLOv8 excelled in detecting objects and activities in complex environments, while the Vision Transformer enhanced the classification capabilities by analysing intricate spatial relationships and extracting meaningful features for weapon identification.

Performance analysis confirmed that the system is capable of handling diverse surveillance scenarios, including varying lighting conditions, complex backgrounds, and crowded scenes. The use of dataset augmentation during training further ensured that the system generalized well across different environments. The robust self-attention mechanism in the Vision Transformer contributed significantly to the accurate classification of weapons, making it a critical component of the system's overall success.

The real-time processing capability, coupled with robust accuracy metrics, made this system a reliable solution for surveillance and security monitoring. By combining YOLOv8 and Vision Transformer, the system provides an effective and scalable solution for suspicious activity detection and classification, ensuring both high performance and robust security in surveillance environments. It can serve as a valuable tool for security personnel to enhance public safety and prevent crime.

7.2 FUTURE ENHANCEMENTS

Scalability and Multi-Object Detection: Future work can focus on improving the system's ability to scale across multiple surveillance cameras, handling large volumes of video data. Enhancing YOLOv8's multi-object detection capabilities and adapting the Vision Transformer for behavior analysis in crowded scenes will be essential for system versatility.

Integration with IoT Devices and Sensors: Incorporating data from additional sensors, such as motion detectors or audio sensors, could help improve the accuracy and robustness of activity detection. These sensors could work in tandem with video surveillance and enhance the Vision Transformer's feature analysis by providing contextual information, such as the sound of weapons being drawn or movements detected in blind spots.

Enhanced Data Privacy and Security: Future enhancements could focus on ensuring that the video and detection data are securely transmitted and stored, addressing privacy concerns while maintaining high levels of detection accuracy. Implementing secure data pipelines and anonymization techniques could safeguard sensitive data while leveraging Vision Transformer's feature extraction without compromising privacy.

Post-Deployment Learning: Enabling the system to continue learning from new video data after deployment will help both YOLOv8 and Vision Transformer adapt to evolving patterns of suspicious activities, improving accuracy over time without requiring complete retraining. Techniques like transfer learning and incremental updates could further enhance the Vision Transformer's capability to classify emerging weapon types and new patterns of suspicious behavior.

REFERENCES

1. Sen, A., Rajakumaran, G., Mahdal, M., Usharani, S., Rajasekharan, V., Vincent, R., &

   Sugavanan, K. (2024). Live Event Detection for People's Safety Using NLP and Deep Learning. ***Journal of Safety and Security Systems****, 45*(3), 123-135.
2. Aziz, A. A. B., & Bajpai, A. (2024). Attire-Based Anomaly Detection in Restricted Areas Using YOLOv8. ***International Journal of Computer Vision and Applications****, 12*(2), 178-190.
3. Sudharson, D., Sirinithi, J., Akshara, S., Abhirami, K., Sriharshita, P., & Priyanka, K. (2024). Proactive Headcount and Suspicious Activity Detection Using YOLOv8. ***IEEE Access****, 11*, 22045-22058.
4. Singh, S., Tyagi, V., Dewangan, S., Krishna, G. S., Reddy, S., & Medi, P. R. (2023). Video Vision Transformers for Violence Detection. ***Computers and Security****, 49*(5), 560-573.
5. Deshpande, K., Sombhadra, S. K., Punn, N. S., & Agarwal, S. (2023). Anomaly Detection in Surveillance Videos Using Transformer-Based Model. ***Journal of Visual Communication and Image Representation****, 85*, 104721.
6. Shoaib, M., Ullah, A., Abbasi, I. A., Algarni, F., & Khan, A. S. (2023). Augmenting the Robustness and Efficiency of Violence Detection Systems for Surveillance and Non-Surveillance Scenarios. ***Applied Intelligence****, 59*(4), 321-339.
7. Mudgal, M., Punj, D., & Pillai, A. (2021). Suspicious Action Detection in Intelligent Surveillance System Using Attribute Modelling. ***Multimedia Tools and Applications****, 80*(16), 23811-23829.
8. Thakur, A., Shrivastav, A., Sharma, R., Kumar, T., & Puri, K. (2024). Real-Time Weapon Detection Using YOLOv8 for Enhanced Safety. ***Journal of Real-Time Image Processing****, 20*(2), 134-149.

APPENDIX

A1-SOURCE CODE

model_test.py

Original file is located at
https://colab.research.google.com/drive/18wRkhrOR8YpaIXKIIy_g07DcKZnxsExg

Dataset Link : https://universe.roboflow.com/le-quy-don-high-school-for-gifted-studentsgfeop/abnormal-activities-u130g print('Tesla T4-GPU') !nvidia-smi

!pip install roboflow from roboflow import Roboflow rf

= Roboflow(api_key="GpxdaRKmGXVQtV3V21ts")

project = rf.workspace("le-quy-don-high-school-for-gifted-studentsgfeop").project("abnormal-activities-u130g") version = project.version(1) dataset = version.download("yolov8")

!pip install ultralytics import ultralytics ultralytics.checks() from ultralytics import YOLO from IPython.display import Image

dataset.location

Train the YOLO model

!yolo task=detect mode=train data={dataset.location}/data.yaml model="yolov8m.pt" epochs=10 imgsz=640

```
Image("/content/runs/detect/train/confusion_matrix.png", width = 600)

Image("/content/runs/detect/train/results.png",width = 600)

Image("/content/runs/detect/train/labels.jpg",width=600)

Image("/content/runs/detect/train/val_batch0_pred.jpg",width=600)

Image("/content/runs/detect/train/val_batch1_pred.jpg",width=600)

Image("/content/runs/detect/train/val_batch2_pred.jpg",width=600)

!yolo task=detect mode=val model="/content/runs/detect/train/weights/best.pt" data={dataset.location}/data.yaml

!yolo task=detect mode=predict model="/content/runs/detect/train/weights/best.pt" conf=0.75

source={dataset.location}/test/images save=True import glob import os from IPython.display

import Image,display for image_path in glob.glob("/content/runs/detect/predict/*.jpg")[:25]:

  display(Image(filename=image_path,width=600))

  print("\n")

!yolo task=detect mode=predict model="/content/runs/detect/train/weights/last.pt" conf=0.75 source="/content/Env_1_out_2X3_mp4-1_jpg.rf.edeabb54d92415b91ddbe6a63a41c85d.jpg" save=True

!yolo task=detect mode=predict model="/content/runs/detect/train/weights/last.pt" conf=0.75 source="/content/4761809-uhd_4096_2160_25fps.mp4" save=True

!yolo task=detect mode=predict model="/content/runs/detect/train/weights/best.pt" conf=0.75 source="/content/fig1.jpeg" save=True

!yolo task=detect mode=predict model="/content/runs/detect/train/weights/last.pt" conf=0.75 source="/content/fig2.mp4" save=True
```

!yolo task=detect mode=predict model="/content/runs/detect/train/weights/best.pt" conf=0.75 source="/content/fig3.mp4" save=True **vit_model.py** import matplotlib.pyplot as plt import torch import torchvision from torch import nn from torchvision import transforms device = "cuda" if torch.cuda.is_available() else "cpu" device

```
# 1. Get pretrained weights for ViT-Base pretrained_vit_weights =
torchvision.models.ViT_B_16_Weights.DEFAULT pretrained_vit_weights

# 2. Setup a ViT model instance with pretrained weights pretrained_vit =
torchvision.models.vit_b_16(weights=pretrained_vit_weights).to(device) pretrained_vit
# Freeze the base params for parameter in
pretrained_vit.parameters():
    parameter.requires_grad = False # 4. Change the classifier head class_names =
['Hammer','Knife','fist'] pretrained_vit.heads = nn.Linear(in_features=768,
out_features=len(class_names)).to(device) pip install torchinfo from torchinfo import
summary
# Print a summary using torchinfo (uncomment for actual output)
summary(model=pretrained_vit,         input_size=(32, 3, 224, 224), #
(batch_size, color_channels, height, width)
        # col_names=["input_size"], # uncomment for smaller output
col_names=["input_size", "output_size", "num_params", "trainable"],
col_width=20,         row_settings=["var_names"]
```

```
)

# Setup directory paths to train and test images train_dir =

'C:/Users/mdkar/Downloads/Project-edit-oct 7/weapon dataset/dataset/train' test_dir =

'C:/Users/mdkar/Downloads/Project-edit-oct 7/weapon dataset/dataset/test'

# Get automatic transforms from pretrained ViT weights

pretrained_vit_transforms = pretrained_vit_weights.transforms()

print(pretrained_vit_transforms) import os from torchvision

import datasets, transforms from torch.utils.data import

DataLoader NUM_WORKERS = os.cpu_count()

def create_dataloaders(

   train_dir:              str,

test_dir: str,

   transform:                  transforms.Compose,

batch_size: int,

   num_workers: int=NUM_WORKERS

):

  # Use ImageFolder to create dataset(s)   train_data =

datasets.ImageFolder(train_dir, transform=transform)   test_data =

datasets.ImageFolder(test_dir, transform=transform)
```

```
 # Get class names   class_names
= train_data.classes   # Turn
images into data loaders
train_dataloader = DataLoader(
    train_data,
batch_size=batch_size,
shuffle=True,
num_workers=num_workers,
pin_memory=True,
 )
 test_dataloader = DataLoader(
    test_data,
batch_size=batch_size,
shuffle=False,
num_workers=num_workers,
pin_memory=True,
 )
 return train_dataloader, test_dataloader, class_names
# Setup dataloaders
train_dataloader_pretrained, test_dataloader_pretrained, class_names =
create_dataloaders(train_dir=train_dir,
```

```
test_dir=test_dir, transform=pretrained_vit_transforms,
batch_size=32) from going_modular.going_modular import engine
from helper_functions import set_seeds # Create optimizer and loss
function optimizer =
torch.optim.Adam(params=pretrained_vit.parameters(),
                 lr=1e-3)
loss_fn = torch.nn.CrossEntropyLoss()
# Train the classifier head of the pretrained ViT feature extractor model
set_seeds() pretrained_vit_results = engine.train(model=pretrained_vit,
train_dataloader=train_dataloader_pretrained,
test_dataloader=test_dataloader_pretrained,
optimizer=optimizer,                    loss_fn=loss_fn,
epochs=10,                    device=device)
# Plot the loss curves from helper_functions
import plot_loss_curves
plot_loss_curves(pretrained_vit_results)

#Test Results import
requests
# Import function to make predictions on images and plot them from
going_modular.going_modular.predictions import pred_and_plot_image
```

```
# Setup custom image path

custom_image_path = "test1.jpg" # Predict on

custom image

pred_and_plot_image(model=pretrained_vit,

image_path=custom_image_path,

class_names=class_names) import requests

# Import function to make predictions on images and plot them from

going_modular.going_modular.predictions import pred_and_plot_image

# Setup custom image path

custom_image_path = "test2.jpg" # Predict on

custom image

pred_and_plot_image(model=pretrained_vit,

image_path=custom_image_path,

class_names=class_names)
```

A2-

SCREENSHOTS

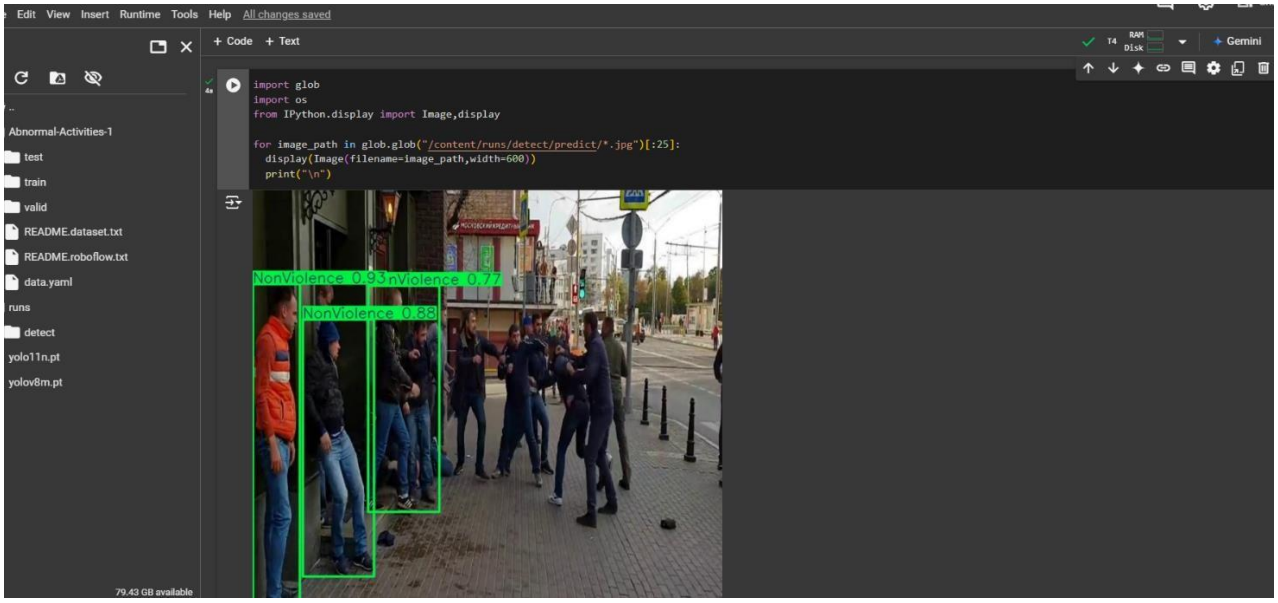

Figure A2.1 Detection of Non-Violence Activity

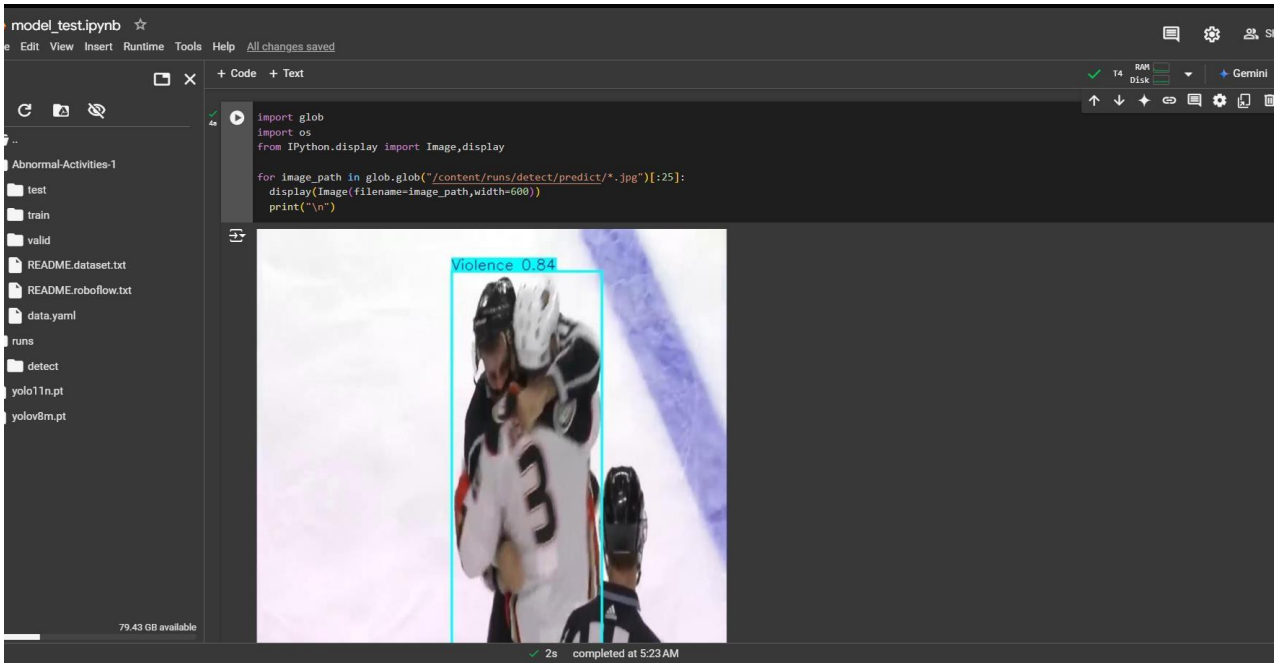

Figure A2.2 Detection of Violence Activity

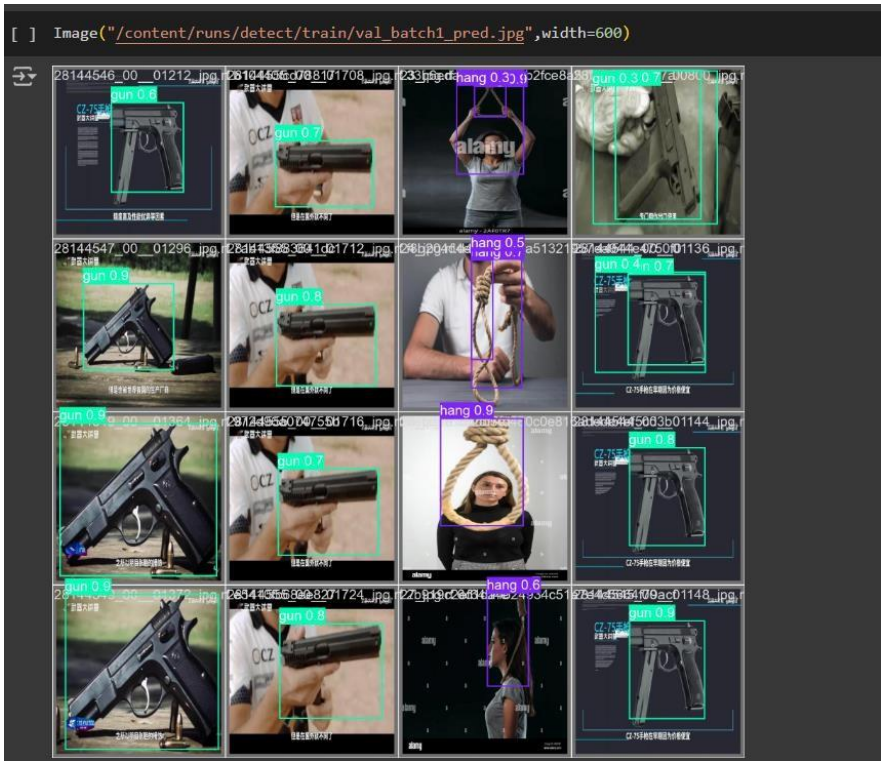

Figure A2.3 Variation of Suspicious activity [Hang and weapons]

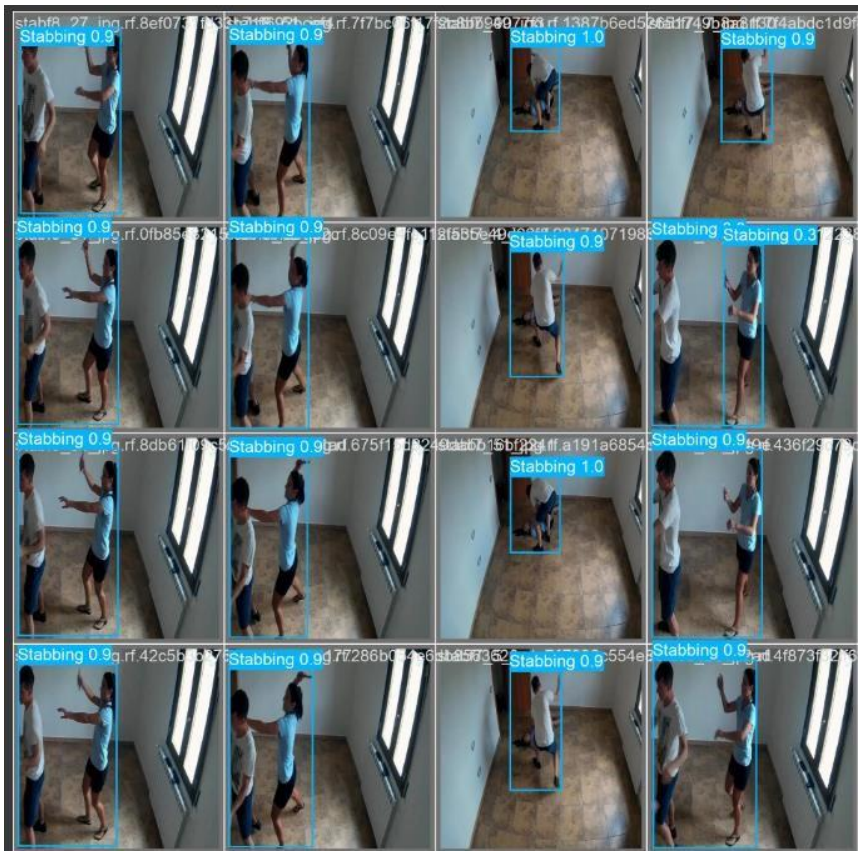

Figure A2.4 Detection of Stabbing activity

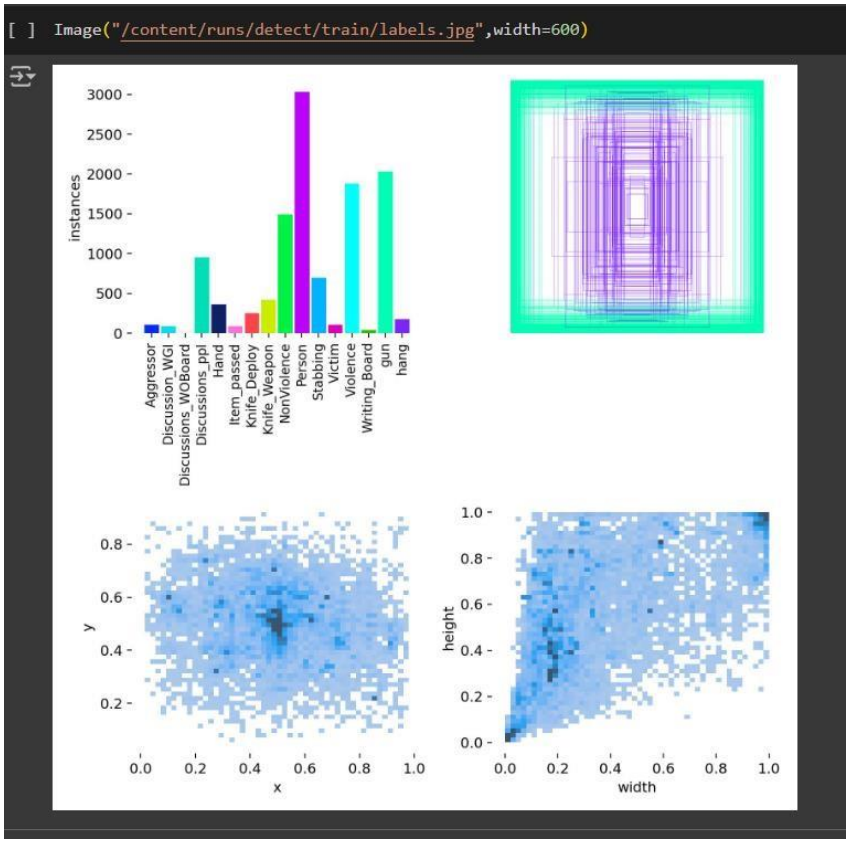

Figure A2.5 Analysis of the classification of suspicious activities

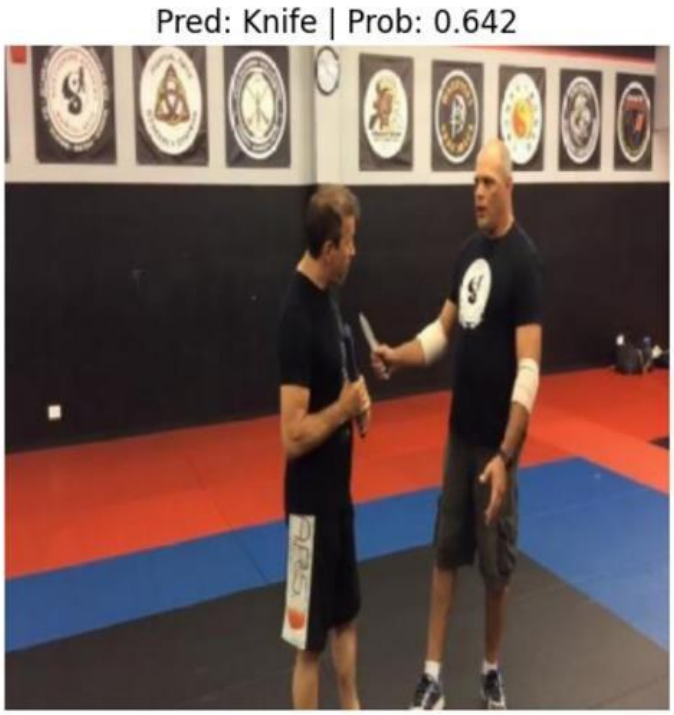

Figure A2.6 Detection of Knife during Violence event

Figure A2.7 Detection of Knife in Suspicious environment

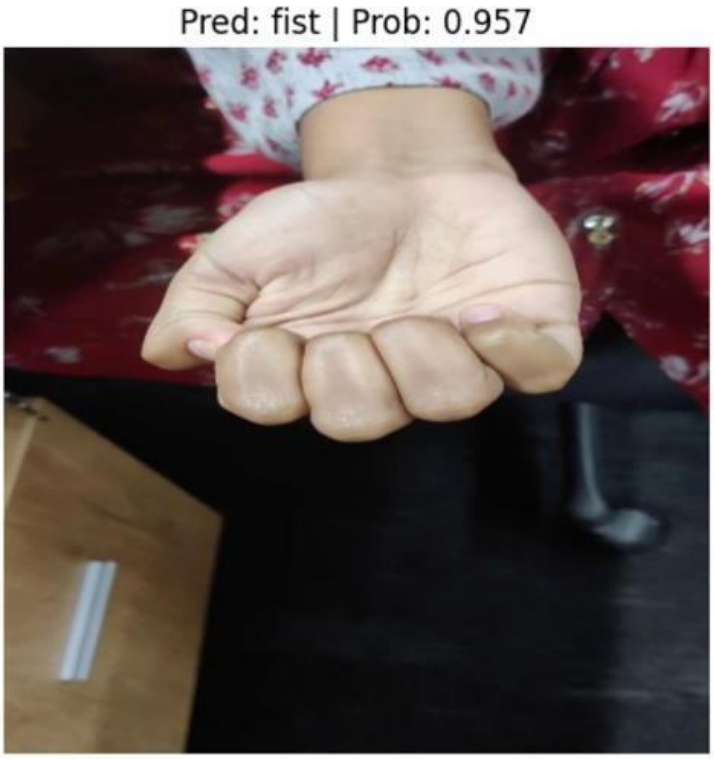

Figure A2.8 Detection of fist movement in violence activity

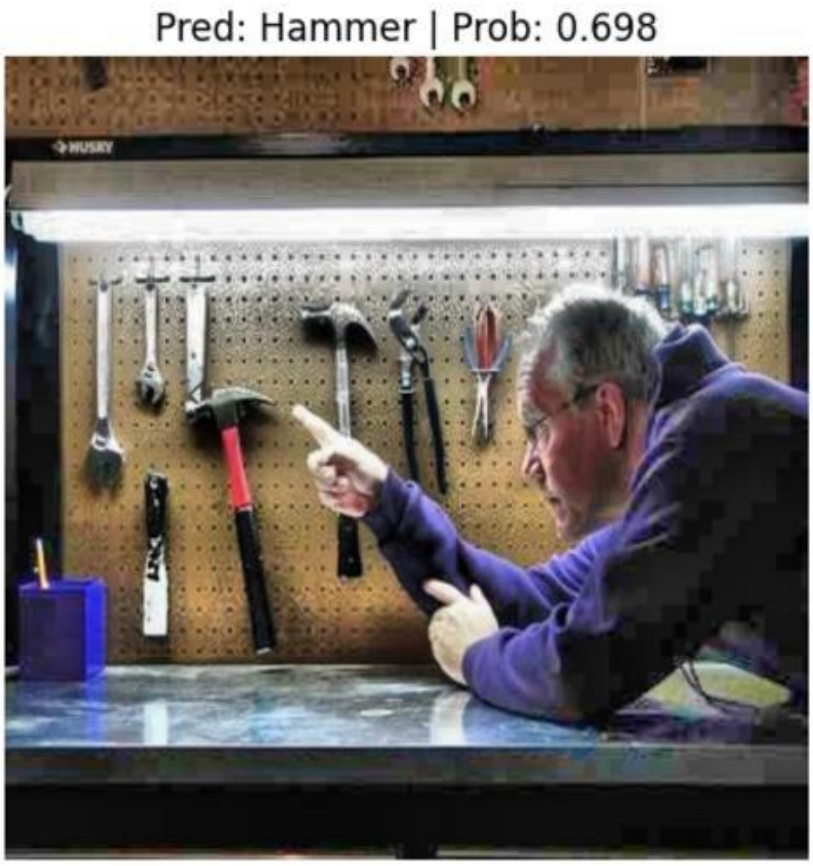

Figure A2.9 Detection of hammer in weapon classification

A3 - TECHNICAL BIOGRAPHY

Mr. MD. Karaamathullah Sheriff, born on December 5, 2001, in Chennai, Tamil Nadu, is an ambitious student currently pursuing an M.Tech degree in Artificial Intelligence and Data Science at B.S. Abdur Rahman Crescent Institute of Science and Technology. With a strong foundation in machine learning and deep learning, Sheriff excels at developing large language model (LLM) applications and implementing advanced AI-driven solutions. His current projects include the Suspicious Activity Tracking System Using YOLOv8 and Vision Transformer, focusing on detecting and classifying suspicious activities like theft and vandalism through real-time video surveillance. This innovative system combines the capabilities of YOLOv8 for object detection and Vision Transformer for weapon classification, ensuring robust security and accurate detection outcomes. He is committed to advancing the field of Artificial Intelligence and creating impactful innovations.