**Violence Detection in Video Streams Using LSTM, GRU, and ANN with Deep Feature Extraction**

A Project Work-II Report

Submitted in partial fulfillment of the requirement of the

Degree of

**BACHELOR OF TECHNOLOGY**

IN

**INFORMATION TECHNOLOGY**

BY

**Raghav Mehrotra [EN21EL301070]  
Piyush Motwani [EN21IT301076]  
Raj Narayan Singh Chouhan [EN21IT301083]**

Under the Guidance of

**Dr. Jyoti Kukade**



**Department of Information Technology**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**JUNE 2025**

**Violence Detection in Video Streams Using LSTM, GRU, and ANN with Deep Feature Extraction**

A Project Work- I Report

Submitted in partial fulfillment of the requirement of the

Degree of

**BACHELOR OF TECHNOLOGY**

IN

**INFORMATION TECHNOLOGY**

BY

**Raghav Mehrotra [EN21EL301070]  
Piyush Motwani [EN21IT301076]  
Raj Narayan Singh Chouhan [EN21IT301083]**

Under the Guidance of

**Dr. Jyoti Kukade**



**Department of Information Technology**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**JUNE 2025**

**Report Approval**

The project work **“Violence Detection in Video Streams Using LSTM, GRU, and ANN with Deep Feature Extraction”** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name: Dr. Jyoti Kukade

Assistant Professor

Medi-Caps University

External Examiner

Name:

Designation:

Affiliation:

**Declaration**

We hereby declare that the project entitled **“Violence Detection in Video Streams Using LSTM, GRU, and ANN with Deep Feature Extraction”** submittedin partial fulfillment for the award of the degree of Bachelor of Technology in ‘Information Technology’ completed under the supervision of **Dr. Jyoti Kukade, Assistant Professor** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Raghav Mehrotra**

**Piyush Motwani**

**Raj Narayan Singh Chouhan**

**Date:**

**Certificate**

I, **Dr. Jyoti Kukade** certify that the project entitled **“Violence Detection in Video Streams Using LSTM, GRU, and ANN with Deep Feature Extraction”** submittedin partial fulfillment for the award of the degree of Bachelor of Technology by **Raghav Mehrotra, Piyush Motwani, Raj Narayan Singh Chouhan** istherecordcarried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Jyoti Kukade

Assistant Professor

Department of Information Technology

Medi-Caps University, Indore

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Prashant Panse

Head of the Department

Department of Information Technology

Medi-Caps University, Indore

**Acknowledgements**

I would like to express my deepest gratitude to the Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Dilip K Patnaik,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm have always boosted my morale. I also thank **Prof. (Dr.) D K Panda,** Pro-Vice Chancellor, Medi-Caps University, **Prof. (Dr.) Pramod S. Nair,** Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Prof. (Dr.) Prashant Panse** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my Internal Guide, **Dr. Jyoti Kukade**, Department of Information Technology, MU, without whose continuous help and support, this project would ever have reached to the completion.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

**Raghav Mehrotra [EN21EL301070]  
Piyush Motwani [EN21IT301076]  
Raj Narayan Singh Chouhan [EN21IT301083]**

B.Tech. IV Year

Department of Information Technology

Faculty of Engineering

Medi-Caps University, Indore

**Abstract**

Time-to-real detection of potential threats has formed the base of maintaining public safety through violence detection systems. The present study dwells on deep learning models that offer improved prediction accuracy for human activity analysis with the changed scope of their AI use.

Recent studies use a fusion of convolutional neural networks (CNNs) and recurrent neural networks (Long Short Term Memory(LSTM)) in many settings to classify violence in videos obtained from surveillance footage, attaining a substantial drop in the rate of classification errors. This research constitutes a hybrid deep learning framework that combines CNNs for extracting features and LSTMs for analyzing violent and non-violent scenes. It was also examined how well the GRUs identified and classified the scenes along with fully connected networks in parallel so as to increase the robustness of the model. CNNs extract spatial features, LSTMs and GRUs capture temporal dynamics. The experiment reaches 92% test accuracy, accomplishing an impressive advance in differentiating violent onset from benign events, achieving superior precision and recall values analyzed with the latest benchmarks. This finding delineates the influence of deep learning in terms of significant enhancement of automated surveillance, paving the path for safer environments through the proactive presence of threat detection.

**Keywords:** LSTM, GRU, ANN, VGG16, MNAS, ROG-AUC

**Table of Contents**

|  |  |  |
| --- | --- | --- |
|  | **Content** | **Page No.** |
|  | Title Page | i |
|  | Report Approval | ii |
|  | Declaration | iii |
|  | Certificate | iv |
|  | Acknowledgement | v |
|  | Abstract | vi |
|  | Table of Contents | vii |
|  | List of figures | ix |
|  | Abbreviations | x |
| Chapter 1 | Introduction |  |
|  | 1.1 Introduction | 1 |
|  | 1.2 Objectives | 2 |
|  | 1.3 Significance | 4 |
|  | 1.4 Problem in Existing System | 6 |
| Chapter 2 | System requirement analysis |  |
|  | 2.1 Information Gathering | 8 |
|  | 2.2 System Feasibility | 9 |
|  | 2.2.1 Economical | 9 |
|  | 2.2.2 Technical | 10 |
|  | 2.2.3 Behavioral | 10 |
|  | 2.3 Platform Specification (Development & Deployment) | 10 |
|  | 2.3.1 Hardware | 10 |
|  | 2.3.2 Technology | 11 |
| Chapter 3 | System Analysis |  |
|  | 3.1 Information Flow Representation | 12 |
|  | |  |  | | --- | --- | | 3.1.1 | Flow Chart Diagram | | 12 |
|  | 3.1.2 Activity Diagram | 13 |
|  | 3.1.3 Data Flow Diagram | 14 |
|  | 3.1.4 Use Case Diagram | 15 |
|  | 3.1.5 Sequence Diagram | 16 |
| Chapter 4 | Design |  |
|  | 4.1 Procedural Approach | 17 |
|  | 4.1.1 Modules Used | 17 |
|  | 4.1.2 Algorithm design for operations | 18 |
|  | 4.2 Interface Design | 20 |
|  | 4.2.1 Interface Design Specification | 20 |
| Chapter 5 | Testing |  |
|  | 5.1 Testing Objective | 21 |
|  | 5.2 Testing Scope | 22 |
|  | 5.3 Testing Principles | 23 |
|  | 5.4 Test Cases and Results | 24 |
| Chapter 6 | Limitations | 27 |
| Chapter 7 | Future Scope | 28 |
| Chapter 8 | Conclusion | 30 |
| Chapter 9 | Bibliography | 31 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| 3.1 | Flow Chart Diagram | 12 |
| 3.2 | Activity Diagram | 13 |
| 3.3 | Data Flow Diagram | 14 |
| 3.4 | Use Case Diagram | 15 |
| 3.5 | Sequence Diagram | 16 |
| 4.1 | UI Design | 20 |
| 4.2 | Output | 20 |

**Abbreviations**

|  |  |
| --- | --- |
| **ABBREVIATION** | **EXPANSION** |
| ANN | Artificial Neural Network |
| LSTM | Long Short-Term Memory |
| GRU | Gated Recurrent Unit |
| MNAS | Mobile Neural Architecture Search |
| ROC - AUC | Receiver Operating Characteristic - Area Under Curve |
| VGG16 | Visual Geometry Group 16-layer network |

**Chapter 1**

**Introduction**

* 1. **Introduction**

Physical violence significantly impacts individuals, families, and societies, affecting mental health and economic stability. Research highlights its alarming prevalence worldwide. A study by Hillis et al. (2016) reported that over half of children globally experienced violence in 2015. The European Union Agency for Fundamental Rights found that one in four Europeans encountered physical violence annually, with 22 million experiencing direct attacks within a year. Given its severity, addressing violence requires a combination of short-term, medium-term, and long-term solutions.

Long-term approaches focus on identifying and preventing the root causes of violence. Studies suggest exposure to aggression in early life contributes to violent tendencies, and factors such as low self-esteem and hostile parent-child relationships play a role. Additionally, technology, particularly violent video games and smartphone use, has been linked to increased aggression.

Medium-term solutions examine the relationship between urban environments and crime. Research using AI, including convolutional neural networks (CNNs) and deep learning models, has explored how population density, green spaces, and urban infrastructure influence crime rates. Studies leveraging Baidu and Google Street View imagery analyse factors like vehicle numbers and pavement conditions, finding correlations with crime patterns.

The most immediate solution is real-time violence detection through AI-powered surveillance systems. Security cameras, combined with advances in big data and AI, facilitate the rapid identification of violent incidents. AI algorithms analyse video footage, recognizing aggressive movements and alerting authorities in real-time. These systems have become increasingly accurate, enhancing public safety while also raising privacy concerns.

Violence detection in videos is a subset of computer vision, specifically action recognition. AI models are trained using annotated datasets containing labelled violent and non-violent activities. Machine learning techniques allow these systems to identify patterns and classify new, unseen videos accordingly. Modern frameworks leverage extensive data processing capabilities to improve detection accuracy.

This research aims to provide a systematic mapping study on AI-driven video violence detection, focusing on physical assault. The primary contribution is an in-depth review of various algorithms, their combinations, and their performance on state-of-the-art datasets. While numerous approaches exist, real-time AI-based detection remains the most effective solution for identifying and mitigating physical violence.

* 1. **Objectives**

1. **Develop an AI-Powered Violence Detection System:**

* This project aims to create an intelligent system that leverages computer vision and deep learning to detect violent activities in video footage automatically. OpenCV will be used to process video frames, extract essential features, and enhance image quality for improved analysis. The system will focus on real-time monitoring to ensure immediate recognition of violent events, helping law enforcement, security agencies, and public safety organizations respond more effectively to potential threats.

1. **Enhance Detection Accuracy Through Advanced AI Models:**

* To maximize accuracy, the system will integrate Convolutional Neural Networks (CNNs) for feature extraction and Recurrent Neural Networks (LSTM, GRU) to model temporal dependencies in video sequences. These deep learning models will be trained and fine-tuned using TensorFlow and Keras, with data augmentation techniques improving their ability to generalize across different environments. By refining model parameters and training strategies, the system will enhance its ability to correctly classify violent and non-violent activities in real-world scenarios.

1. **Ensure Real-Time Detection with Minimal Latency:**
   * Real-time analysis of high-resolution images for accurate weed detection requires robust image processing methods. Advanced edge detection, morphological filtering, and segmentation methods enhance the resolution and focus of images for the system to function well even under different environmental conditions such as lighting, weather, and soil texture for proper output in various scenarios.
2. **Achieve High Classification Accuracy and Reliability:**

* Accurate classification is critical to minimizing false positives and false negatives. By employing CNNs, LSTM, and GRU models, the system will differentiate between violent and non-violent events with high precision. Extensive training on diverse datasets will ensure the model is robust against variations in lighting, camera angles, and movement patterns. Performance metrics such as precision, recall, and F1-score will be used to fine-tune the system, ensuring it delivers reliable results under real-world conditions.

1. **Optimize Performance for Scalability Across Multiple Video Streams:**

* To handle large-scale surveillance applications, the system must be capable of processing multiple video streams simultaneously. Optimization techniques like model quantization, pruning, and batch processing will be employed to reduce computational overhead while maintaining accuracy. The system will be designed for deployment on cloud platforms, edge devices, or on-premise servers, allowing flexibility for different security applications.

1. **Validate System Performance Through Field Testing:**

* Comprehensive field trials will be conducted to assess the system’s accuracy, responsiveness, and adaptability to various environments. Real-world testing will help identify challenges such as occlusions, lighting variations, and camera positioning. Feedback from security agencies and law enforcement will be used to refine the system, improving its overall efficiency and effectiveness. These trials will provide critical insights that ensure the system meets practical needs and can operate reliably in live surveillance scenarios.

1. **Promote Adoption and Integration into Security Systems:**

* To maximize its impact, the system must be easy to integrate with existing surveillance infrastructure. Collaborations with law enforcement, security firms, and government agencies will facilitate adoption. Workshops, demonstrations, and case studies will be used to showcase its benefits, emphasizing cost-effectiveness, ease of deployment, and improved public safety. Clear documentation and user-friendly interfaces will further encourage widespread use, making AI-powered video violence detection a standard tool in modern security systems.

1. **Ensure Ethical Considerations and Privacy Protection:**

* While AI-driven surveillance enhances security, it raises ethical and privacy concerns. The system will implement strict data protection measures to prevent misuse and unauthorized access. Ethical guidelines will be developed to ensure responsible deployment, balancing security needs with individual privacy rights. Transparency in AI decision-making, along with compliance with legal frameworks, will help build trust in the technology and mitigate concerns regarding mass surveillance and potential biases in violence detection algorithms.

1. **Support Continuous Research and Development:**

* As AI and deep learning technologies evolve, the system must be continuously improved to stay ahead of emerging security threats. Future enhancements will explore predictive analytics, multi-camera integration, and behavioral analysis for more comprehensive violence detection. Research efforts will focus on refining AI models, increasing computational efficiency, and expanding dataset diversity to improve detection capabilities.

* 1. **Significance**

1. **Automating Violence Detection with AI:**

* The proposed system leverages advanced computer vision and deep learning to detect violent activities in real-time. By reducing the need for manual surveillance, it enhances security and public safety. Utilizing OpenCV and deep learning models, the system can analyze video footage with high precision, distinguishing violent incidents from normal activities. This automation reduces human errors and ensures a proactive approach to crime prevention, making it a critical tool for law enforcement and security agencies.

1. **Enhancing Detection Accuracy with Advanced Algorithms:**

* Deep learning techniques, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), improve the system’s ability to detect violent actions accurately. These models learn from extensive datasets and recognize subtle movement patterns that indicate aggression. Compared to traditional surveillance methods, AI-based detection significantly reduces false positives and negatives, ensuring a reliable and robust system for identifying violence in crowded and complex environments.

1. **Optimizing Image and Video Processing for Real-Time Analysis:**

* For effective real-time violence detection, the system employs high-performance image processing techniques. Methods such as motion detection, object tracking, and spatiotemporal analysis ensure accurate identification of violent behavior. Advanced filtering and segmentation techniques help the system adapt to different lighting conditions, camera angles, and environmental factors. This ensures consistent accuracy, whether the footage is taken during the day, at night, or in dynamic urban settings.

1. **Reducing Response Time to Violent Incidents:**

* By providing instant alerts upon detecting violent activity, the system enables security personnel to respond quickly to threats. The integration of AI-powered surveillance minimizes delays in identifying and addressing violent events. Real-time processing ensures that incidents are flagged immediately, allowing law enforcement agencies and emergency responders to intervene swiftly, preventing further harm. This capability is crucial in high-risk areas such as public spaces, schools, and transportation hubs.

1. **Improving Public Safety and Crime Prevention:**

* An AI-driven violence detection system acts as both a deterrent and an early warning mechanism. By analyzing live video feeds, it can identify escalating situations before they turn into violent encounters. The system can also assist in forensic analysis by storing and categorizing violent event data, helping authorities in criminal investigations.

1. **Overcoming Challenges of Manual Surveillance:**

* Manual monitoring of surveillance footage is labor-intensive, time-consuming, and prone to human error. The AI-based violence detection system automates this process, reducing the burden on security personnel while improving efficiency. This not only lowers operational costs but also ensures that violent activities are detected with high accuracy. The system’s continuous learning capabilities allow it to adapt and improve over time, further enhancing its effectiveness.

1. **Validating the System Through Real-World Testing:**

* Extensive field trials will be conducted to evaluate the system’s performance in real-world conditions. These tests will assess its accuracy, response time, and adaptability to different environments. Feedback from law enforcement agencies, security experts, and end-users will be incorporated to refine the system. Continuous improvements based on real-world data will ensure that the technology remains effective and reliable in practical applications.

* 1. **Problem in Existing System**

1. **Inefficiency of Manual Surveillance:**

* Traditional security systems rely heavily on manual monitoring by human operators, which is prone to fatigue and errors. Security personnel often miss critical events due to long hours of video observation. Additionally, the inability to monitor multiple camera feeds simultaneously reduces the overall effectiveness of crime prevention and response. This inefficiency results in delayed action against violent incidents, increasing risks to public safety.

1. **High False Alarm Rates in Conventional Systems:**

* Existing automated surveillance systems often generate high false positives and negatives when detecting violent behavior. Simple motion detection or rule-based video analysis struggles to differentiate between normal activities (e.g., sports, sudden movements) and actual violence. As a result, security teams either respond unnecessarily to false alarms or miss real violent incidents, leading to compromised safety in high-risk areas.

1. **Lack of Advanced AI Integration:**

* Most existing security cameras rely on basic motion detection and object tracking, which are not sufficient for identifying complex human behaviors like aggression and violence. Without the integration of deep learning models such as CNNs, LSTMs, and GRUs, these systems cannot accurately analyze violent activities. The lack of AI-driven solutions limits the ability to recognize subtle yet critical patterns that indicate potential threats.

1. **Privacy and Ethical Concerns in Surveillance:**

* Mass surveillance using traditional cameras raises privacy concerns, as they often record individuals continuously without distinguishing between normal and violent activities. Many existing systems do not include privacy-preserving measures such as anonymization or selective recording. This can lead to ethical and legal issues, making widespread adoption challenging without proper safeguards.

1. **Scalability Challenges in Large Surveillance Networks:**

* Conventional video monitoring systems face difficulties in scaling across multiple locations and environments. Processing video feeds from multiple cameras in real-time requires significant computational power, which traditional systems lack. Without optimization techniques such as cloud-based AI processing or edge computing, existing surveillance networks struggle to handle large-scale deployments efficiently.

1. **Delayed Incident Reporting and Response:**

* Since most traditional surveillance systems do not have automated alert mechanisms, security teams often discover violent incidents only after reviewing recorded footage. This delay prevents immediate intervention, allowing crimes to escalate or offenders to escape. An effective violence detection system should provide instant notifications to authorities, ensuring timely action and potentially saving lives.

1. **Environmental Limitations Affecting Detection Accuracy:**

* Existing surveillance systems struggle with changing environmental conditions such as poor lighting, occlusions, and crowded spaces. Traditional cameras fail to detect violent behavior accurately in low-light conditions or when the field of view is obstructed. AI-based solutions with advanced image processing techniques could overcome these challenges, ensuring higher detection accuracy in diverse real-world scenarios.

1. **Ineffective Integration with Modern Security Infrastructure:**

* Many current security systems operate in isolation and do not integrate with law enforcement databases, emergency response systems, or smart city infrastructure. This lack of connectivity reduces their effectiveness in preventing crime. An AI-driven surveillance solution should seamlessly integrate with modern security frameworks, enabling coordinated responses, automated reporting, and enhanced crime analytics.

**Chapter 2**

**System Requirement Analysis**

**2.1 Information Gathering**

**a) Dataset Utilization:**

* The Real-Life Violence Situations Dataset from Kaggle was selected for its extensive collection of annotated video frames featuring violent and non-violent activities. This dataset provides diverse real-world conditions, including variations in lighting, crowd density, and motion dynamics. To enhance the model’s learning capabilities, preprocessing techniques such as frame extraction, resizing, normalization, and data augmentation were applied. These steps ensure that the model generalizes well across various surveillance environments, improving its reliability in real-time violence detection.

**b) Model Selection and Training:**

* To achieve robust violence detection, Artificial Neural Networks (ANNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) were implemented. ANNs were utilized for initial feature extraction, while LSTM and GRU networks captured temporal dependencies in video sequences, ensuring accurate recognition of violent activities over time. The model underwent iterative training using optimized hyperparameters, enhancing its ability to differentiate between violent and non-violent behavior with high precision and recall.

**c) Computer Vision Techniques:**

* Advanced image and video processing techniques were employed to improve detection accuracy. Keyframe extraction methods ensured that relevant frames were processed while filtering out redundant information. Data augmentation techniques such as flipping, rotation, and brightness normalization were applied to enhance the dataset’s variability. By leveraging optical flow analysis and motion tracking, the model effectively recognized aggressive movements, distinguishing them from normal human interactions.

**d) Limitations of Traditional Violence Detection:**

* Traditional surveillance methods rely on manual monitoring and simple motion detection algorithms, which suffer from inefficiencies such as human fatigue, subjectivity, and high false alarm rates. Motion-based techniques struggle to differentiate between everyday movements and violent actions, leading to misclassifications. These limitations emphasize the necessity of AI-driven violence detection models that can analyze video sequences in real time and provide accurate alerts for security personnel.

**e) Scalability and Performance Optimization:**

* The system was designed for scalability and real-time performance, making it suitable for deployment across various environments, including public surveillance, law enforcement, and private security systems. Optimization techniques such as model pruning, quantization, and batch processing were implemented to improve computational efficiency. The system was designed to function on cloud-based platforms, edge devices, and on-premise servers, ensuring real-time violence detection without performance degradation.

**2.2 System Feasibility**

**2.2.1 Economical:**

* The proposed AI-driven violence detection system offers significant economic benefits by reducing reliance on human surveillance and minimizing response times for security threats. Traditional monitoring requires continuous human oversight, leading to high labor costs and potential inefficiencies. By leveraging deep learning models such as ANN, LSTM, and GRU, the system automates real-time analysis, reducing the need for extensive manpower. Additionally, its lightweight design ensures compatibility with existing surveillance infrastructure, eliminating the need for costly hardware upgrades. The system's ability to prevent incidents proactively can save organizations from financial losses due to violence-related damages and liabilities.

**2.2.2 Technical:**

* Technically, the system integrates advanced artificial neural networks to extract meaningful features from video frames, while LSTM and GRU networks capture temporal patterns to accurately distinguish between violent and non-violent events. High-resolution video processing ensures precise detection even in low-light or crowded environments. The model supports real-time deployment on cloud servers, edge devices, and on-premise security systems, allowing seamless integration into existing surveillance networks. Optimization techniques such as model quantization and pruning enhance computational efficiency, ensuring minimal latency in real-time threat detection.

**2.2.3 Behavioral:**

* The system promotes a behavioral shift in security and law enforcement by introducing AI-powered automation into violence detection. Security personnel can focus on critical decision-making rather than manually monitoring multiple video feeds. The user-friendly interface allows easy adoption, even for non-technical users, while automated alerts ensure swift incident responses. By improving safety measures and reducing false alarms, the system fosters trust among users, encouraging widespread acceptance in public spaces, workplaces, and law enforcement agencies. Its potential to prevent violence before escalation makes it a crucial tool for modern security management.

**2.3 Platform Specification (Development & Deployment)**

**2.3.1 Hardware:**

**a) Development Environment:**

* Processor: High-performance multi-core CPU (Intel i5 or equivalent) for efficient data processing and model training.
* RAM: Minimum of 32 GB to handle large datasets and intensive computations during model training and inference.
* GPU: NVIDIA RTX 2080 Ti or equivalent for accelerated deep learning tasks, such as training neural networks.
* Storage: SSD with at least 1 TB capacity for fast access to datasets and model storage.
* Operating System: Windows 10 or Linux-based system (Ubuntu preferred) with support for Python, TensorFlow, and PyTorch.

**b) Deployment Environment:**

* Processor: Quad-core CPU (Intel i5 or equivalent) with sufficient clock speed for live inference applications.
* RAM: At least 16 GB for handling image segmentation and real-time analysis.
* GPU: NVIDIA T4 or equivalent for on-the-edge computing or cloud-based inference.
* Storage: On-premises storage with 500 GB SSD for storing processed data and results.
* Network: High-speed internet connection with a minimum of 50 Mbps for data upload/download.
* Operating System: Ubuntu or Windows Server for deployment of the model.

**2.3.2 Technology:**

**a) Development Platform:**

* Programming Language: Python (primary language for deep learning, data analysis, and real-time video processing).
* Deep Learning Frameworks:
  + TensorFlow/Keras: Used for training and deploying ANN, LSTM, and GRU models for sequential video analysis.
  + PyTorch: Alternative framework for fine-tuning models and optimizing performance.
* Libraries:
  + OpenCV: For image preprocessing and handling real-time video feeds.
  + NumPy/Pandas: For data manipulation and analysis.
  + Scikit-learn: For evaluation and additional machine learning techniques.
* IDE: Jupyter Notebook or Visual Studio Code for seamless development and debugging.
* Version Control: Git and GitHub/GitLab for managing model versions and collaborative development.

**b) Deployment Platform:**

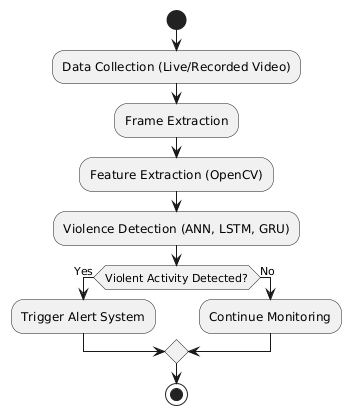
* Cloud Platforms:
  + - Google Cloud Platform (GCP) or AWS: For deploying trained ANN, LSTM, and GRU models, enabling real-time violence detection and scalability.
* Web Technologies:
  + - Flask or FastAPI: To build a RESTful API for real-time interaction with the deployed model.
* Data Storage:
* On-premises SSD storage (500 GB) or cloud storage solutions (Google Drive, AWS S3) for storing processed video data, model outputs, and logs.

**Chapter 3**

**System Analysis**

**3.1 Information Flow Representation**

**3.1.1 Flow Chart Diagram:**

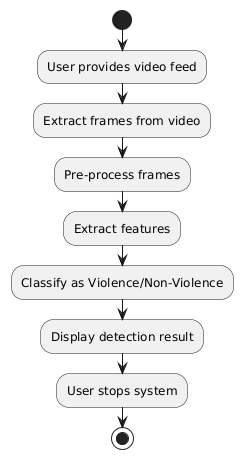


**Figure 3.1: Flowchart diagram**

**Description:**

1. Start: Initialize the system for real-time violence detection.
2. Data Collection: Capture live or recorded video footage for analysis.
3. Frame Extraction: Convert video into individual image frames for processing.
4. Feature Extraction: Apply OpenCV to extract key features such as motion, object detection, and body posture.
5. Violence Detection: Utilize ANN, LSTM, and GRU models to analyze temporal patterns in the extracted frames.
6. If no violence is detected, the system loops back to continue monitoring.
7. If violence is detected, proceed to the next step.
8. Trigger Alert System: Activate an alert mechanism (e.g., notifications, sirens, or emergency response).
9. Stop: End the process or reset for continued monitoring.

**3.1.2 Activity Diagram:**

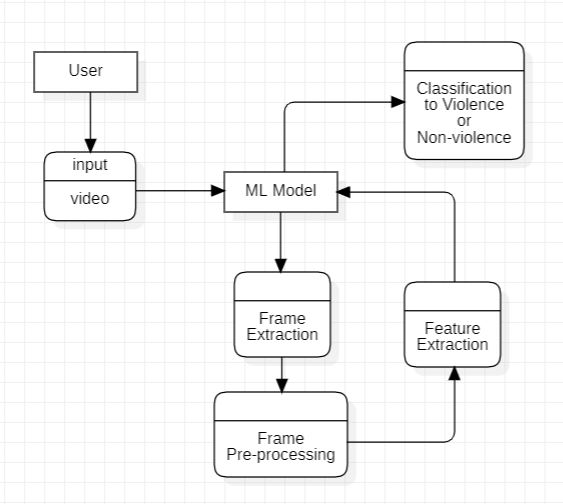


**Figure 3.2: Activity diagram**

**Description:**

* + 1. Start System: Initialize the system for real-time violence detection.
    2. User Provides Video Feed: The system receives a live or recorded video for analysis.
    3. Extract Frames from Video: Convert the video into individual frames for further processing.
    4. Pre-process Frames: Apply filtering, resizing, and normalization to enhance the quality of the frames.
    5. Extract Features: Use OpenCV and deep learning techniques to extract motion-based and spatial features.
    6. Classify as Violence/Non-Violence: Apply ANN, LSTM, and GRU models to classify the extracted features.
    7. Display Detection Result: Show the classification outcome to the user.
    8. User Stops System: The user terminates the system or allows continuous monitoring.
    9. Stop: End the process after detection and decision-making.

**3.1.3 Data Flow Diagram:**

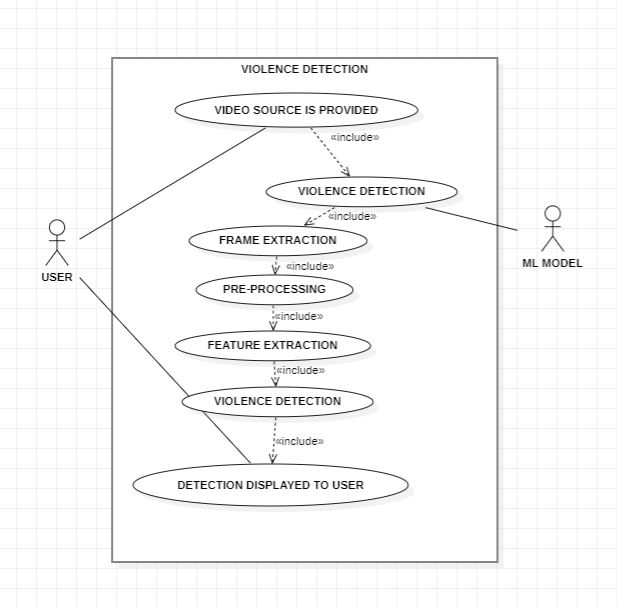


**Figure 3.3: Data Flow diagram**

**Description:**

* 1. User Inputs Video: The user provides a video feed to the system for analysis.
  2. ML Model Processes Input: The video is sent to the ML model for further processing.
  3. Frame Extraction & Pre-processing: The model extracts frames and applies necessary enhancements.
  4. Feature Extraction & Classification: Extracted features help classify the video as Violence or Non-Violence.
  5. Output is Delivered to User: The classification result is displayed to the user.

**3.1.4 Use Case Diagram:**



**Figure 3.4: Use-Case diagram**

**Description:**

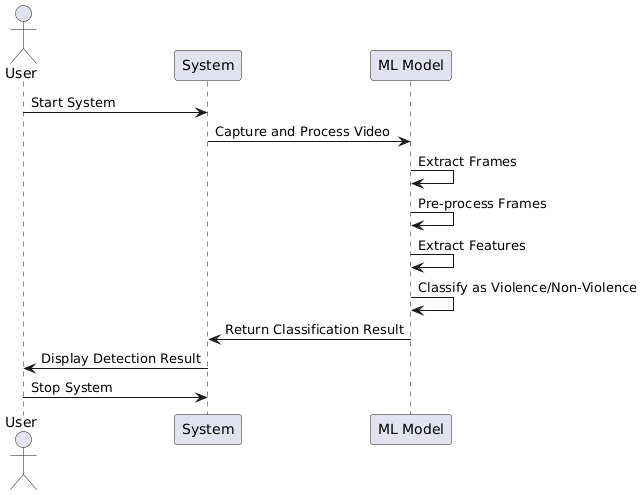
Key Actors and Use Cases:

* Operator: Initiates, monitors, and terminates the system.
* ANN, LSTM, GRU Model: A deep learning framework used for feature extraction and classification of violent and non-violent activities.

Core Use Cases:

1. Start/Stop System: The operator initiates and terminates the system's operation.
2. Capture Video Frames: The system captures live surveillance footage or processes recorded videos.
3. Feature Extraction: The system extracts spatial and temporal features using computer vision techniques and deep learning frameworks like OpenCV and TensorFlow.
4. Violence Detection: The deep learning model (ANN, LSTM, GRU) analyzes the extracted features and classifies the scene as violent or non-violent.
5. Display Detection Results: The classified results (violence detected or not) are displayed to the user in real-time.

**3.1.5 Sequence Diagram:**



**Figure 3.5: Sequence diagram**

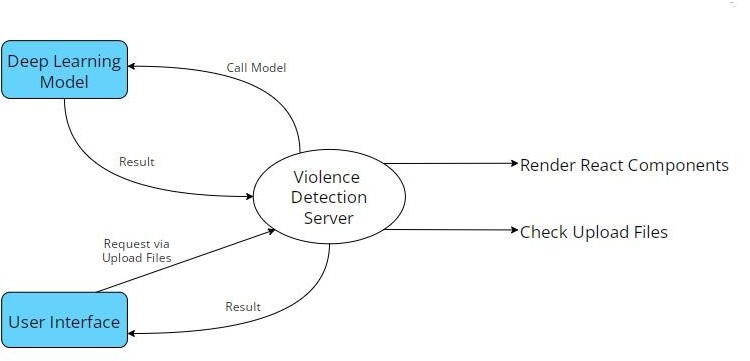
**Description:**

1. User starts the system: The user initiates the violence detection system, activating all processes.
2. Video is provided: The system receives a live or recorded video feed for analysis.
3. ML Model extracts frames: The video is broken down into individual frames for processing.
4. ML Model pre-processes frames: The extracted frames undergo enhancement techniques such as resizing, noise reduction, and normalization to improve analysis accuracy.
5. ML Model extracts features: The system identifies key spatial and temporal features from the video frames that help distinguish violent and non-violent actions.
6. ML Model classifies the event: Using ANN, LSTM, and GRU, the system determines whether the event in the video is violent or non-violent.
7. System returns classification result: The system processes the model’s output and shows it for display.

**Chapter 4**

**Design**

**4.1 Architectural Design**

**4.1.1 Architectural Context Diagram:**

**Fig 4.1 Architectural Context Diagram**

**4.1.2 Description of Architectural Diagram:**

* User or Applicant hits the post request and upload video file, then Flask backend calls the inference deep learning model from Server and get the predicted output from it.
* Then backend renders the resultant React Components to the user with desire output.
* Model launches the inference of video classification form getting the prediction via it.
* User or Applicant sees the prediction on its interface with confidence.

1. **Weed Classification:**

* The detected weeds are analyzed using a pre-trained model to determine their type, such as broadleaf or grass, and their severity level, categorized as low, medium, or high. The classification results are recorded to guide targeted intervention strategies.

1. **Decision Support:**

* Based on the classification data, the system evaluates each detected weed's severity and type to generate appropriate actions. It creates precise spraying instructions, including the target GPS coordinates, ensuring resource-efficient and accurate weed management.

1. **Herbicide Spraying:**

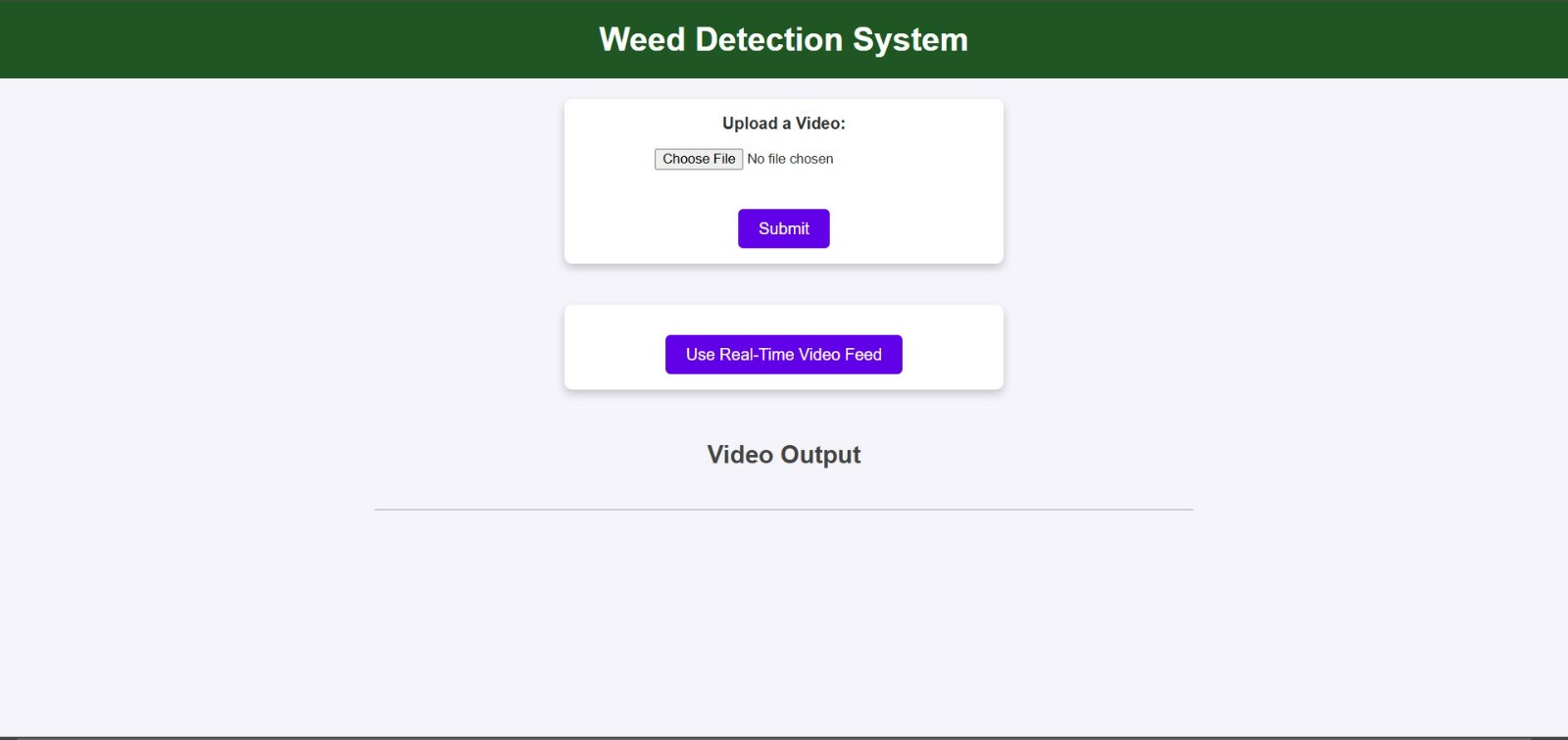
* Following the generated instructions, the drone navigates to the designated GPS locations and activates the herbicide dispenser for targeted spraying. Each successful action is logged, enabling detailed tracking of operations and ensuring accountability.

1. **Return to Base:**

* Upon completing the operation, the drone autonomously returns to the base. Comprehensive logs of detected weeds and the actions performed are stored for analysis and reporting, facilitating continuous improvement of the system.

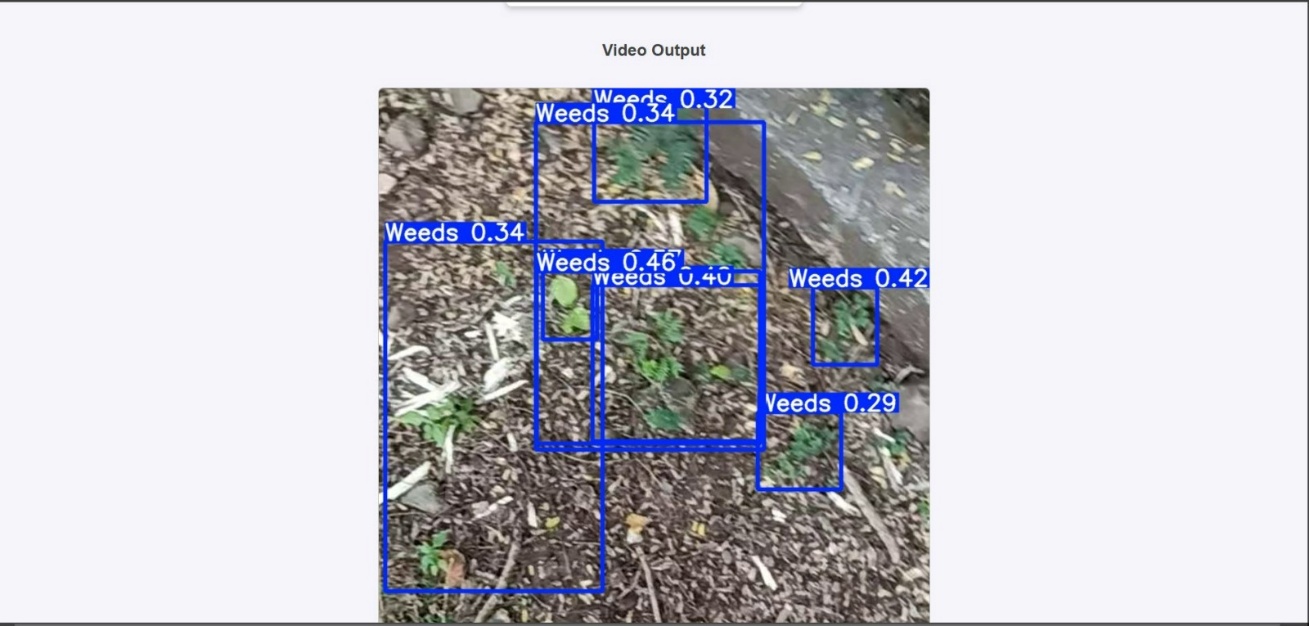
**4.2 Interface Design**

**4.2.1 Interface Design Specification:**



**Figure 4.1: UI Design**

Figure 4.1, depicts the User Interface in which the user can upload a video file chosen from his local machine or he can go for the real-time video streaming by allowing the device camera to be accessed and facilitating the weed detection.



**Figure 4.2: Output**

Figure 4.2, represents the output that will be appeared on the particular device upon allowing it for weed detection through that particular machine.

**Chapter 5**

**Testing**

**5.1 Testing Objective**

1. **Accuracy Testing:**

* The weed detection and classification model, YOLOv8, is thoroughly tested to ensure that it accurately identifies and classifies weeds under different environmental conditions. This includes testing under different lighting, crop types, and weed densities to confirm reliability and adaptability in real-world scenarios.

1. **Performance Testing:**

* System performance is validated by establishing its ability to process frames of video in an effective manner, detect weeds in a real-time environment, and minimize latency during decision and herbicide spraying. These ensure that the system remains highly effective and responsive during operational activities.

1. **Integration Testing:**

* All the components of the system, including the weed-detection module, decision-support system, and hardware modules for control, are assessed for smooth communication and integration. Proper data flow between such modules ensures smooth and coordinated activity.

1. **Functionality Testing:**

* The system is tested comprehensively to ensure that it carries out all the intended operations, including video capture, weed detection, classification, GPS mapping, and herbicide application. This broad testing ensures the integrity of the system in terms of functionality.

1. **Robustness Testing:**

* The system is tested in terms of its ability to perform under challenging conditions, such as overlapping weeds, blurred images, or incomplete farm scans. Recovery mechanisms are tested to ensure that the system remains effective and resilient in unexpected conditions.

1. **Scalability Testing:**

* The system is tested on scalability by simulating larger farm areas and higher weed densities. It tests the system so that it will verify the performance and accuracy while dealing with increased workloads and complex scenarios.

1. **User Interface Testing:**

* The user interface is tested to ensure that it is user-friendly for farmers and operators. Clear instructions, easy configuration options, and detailed reporting features are validated to ensure usability without requiring advanced technical knowledge.

1. **Safety Testing:**

* Safety protocols are validated to confirm that the system avoids non-weed regions, protects crops, and prevents excessive herbicide application. This testing ensures that the system operates responsibly and aligns with environmental safety standards.

1. **Report Validation:**

* The accuracy of the reports is verified by comparing the weed counts detected, classified, and treated with GPS-based location details. This ensures the reports offer an accurate and truthful representation of system operations.

**5.2 Testing Scope**

1. **Weed Detection and Classification Accuracy:**
   * It rigorously tests for its capability to detect weed types in the environment of all shapes, sizes, colors, etc. Testing has been conducted on different lighting, weather conditions, as well as various crop growth stages. Therefore, there is a good chance it could be strong enough to show real-time performances in any agricultural environments.
2. **System Performance:**

* Real-time video frame processing is tested to prove the effectiveness of weed detection and classification. The system's capacity to work flawlessly under high workloads, such as dense weed populations or large farm areas, is tested to confirm its reliability and responsiveness in demanding conditions.

1. **Component Integration:**

* The interaction of various modules is validated to ensure the smooth flow of data and functionality. Components such as the weed detection system, decision-making engine, and control modules for the execution of actions are tested for synchronized operation, with outputs from one module being checked for proper interpretation by the next.

1. **Hardware Compatibility:**

* There must be a test of its ability to work with every part of hardware, including cameras, processing units, and control systems within both development and deployment environments; otherwise, the intended configuration of hardware will ensure the system operates smoothly and effectively.

1. **User Interface Functionality:**

* The user interface is tested to establish that it is intuitive, simple to navigate, and will work on various devices. Setup options for input parameters and the clarity and accessibility of the output reports are evaluated for the operators to be able to manage and interpret the workings of the system efficiently.

1. **Environmental Adaptability:**

* The adaptability of the system to real-world environmental changes such as variations in terrain, weather, and crop arrangement is tested. This would test how the system remains functional and efficient under the changing nature of agricultural fields.

1. **Error Handling and Recovery:**

* The system is tested for its tolerance to unexpected errors like incomplete video feeds, weed detection failure, or even hardware malfunctions. Its autonomous self-recovery capacity and ability to resume operations with minimal human interruption after such incidents are evaluated.

1. **Scalability and Extensibility:**

* To test the scalability of the system, operations will be simulated in larger agricultural fields with higher weed densities. Extensibility will also be validated through the inclusion of new categories of weeds and integration of potential updates to hardware or software for the system.

**5.3 Testing Principles**

1. **Test Early and Often:**

* Testing begins early in the development process, and defects are detected as soon as possible. It helps maintain the system functionality throughout the cycle and saves a lot of trouble during later stages of the process.

1. **Defect Prevention:**

* Preventing defects is more efficient than detecting and fixing them. By adhering to best practices in design, conducting thorough code reviews, and carefully analyzing requirements, the likelihood of introducing defects into the system is significantly reduced.

1. **Exhaustive Testing is Not Possible:**

* Although thorough testing is needed, it is impossible to test every possible input or scenario. It should rather be focused on critical functionalities, edge cases, and typical user behaviors so the system can be expected to function reliably in real conditions.

1. **Testing Should be Based on Requirements and Specifications:**

* The testing process must closely follow the documented requirements and specifications of the system. In this way, all functionalities, performance, and usability metrics will be properly tested against expectations.

1. **Early Defect Detection Saves Cost:**

* The earlier defects are identified and resolved, the less it will cost. By using practices such as continuous integration and automated testing, issues can be detected and resolved before they escalate into larger problems.

1. **Testing for Verification and Not Just Breakage:**

* Testing is mainly done to validate that a system operates the way it is intended. Verification therefore ensures conformity with design as well as functional requirements to ensure that overall quality assurance is delivered.

1. **Independence of Testers:**

* Independent testers who are not involved in the development process bring an unbiased perspective. Their detachment from development tasks enables them to identify issues that the developers might have overlooked, ensuring a more comprehensive testing process.

1. **Repeatable Testing:**

* Test cases should be designed to be repeatable under identical conditions. Consistent results allow teams to verify fixes, measure improvements, and maintain quality assurance throughout the development lifecycle.

**5.4 Test Cases and Results**

**a) Test Case 1: Video Capture and Frame Extraction**

* **Objective**: Ensure the drone system correctly captures video and extracts frames for analysis.
* **Preconditions**: Drone is in operation mode, and the camera is functional.
* **Test Steps**:
  1. Start the drone system and activate the camera.
  2. Capture a video of a farm area.
  3. Extract frames from the captured video.
  4. Verify that the frames are extracted at regular intervals (e.g., every second).
* **Expected Result**: Video frames are extracted successfully and stored for further processing.
* **Pass/Fail Criteria**: The system should extract frames without errors, and the video quality should be sufficient for weed detection.
  1. **Test Case 2: Weed Detection using YOLOv8**
* **Objective**: Ensure YOLOv8 correctly detects weeds in the captured frames.
* **Preconditions**: Frames from video have been extracted, and YOLOv8 is trained for weed detection.
* **Test Steps**:
  1. Send extracted frames to the YOLOv8 model.
  2. Process the frames for weed detection.
  3. Check if the detected objects are correctly classified as weeds.
  4. Verify that the detection results are marked with bounding boxes around identified weeds.
* **Expected Result**: YOLOv8 should accurately classify weeds and draw bounding boxes around them in the frames.
* **Pass/Fail Criteria**: The system should correctly identify weeds in at least 90% of the cases based on ground truth data.

**c) Test Case 3: Herbicide Dispensing Trigger**

* **Objective**: Ensure the herbicide dispenser is triggered when weeds are detected.
* **Preconditions**: The drone is airborne, and weeds are detected.
* **Test Steps**:
  1. Detect weeds in the captured frames using the weed detection model.
  2. Trigger the herbicide dispensing system when weeds are detected.
  3. Verify that the herbicide dispenser is activated.
  4. Ensure the drone moves to the weed location before dispensing herbicide.
* **Expected Result**: The herbicide dispenser should be triggered only when weeds are detected, and the drone should spray herbicide at the correct location.
* **Pass/Fail Criteria**: The herbicide dispenser should be activated and function properly with no false triggers.

**d) Test Case 4: Drone Navigation and Weed Location Marking**

* **Objective**: Ensure the drone navigates to the correct location and marks weed positions.
* **Preconditions**: Weed locations have been detected by the YOLOv8 model.
* **Test Steps**:
  1. Mark the weed location detected by the model.
  2. Direct the drone to fly to the marked weed location.
  3. Verify that the drone successfully navigates to the location and marks the position.
  4. Ensure the system logs the weed treatment results.
* **Expected Result**: The drone should correctly navigate to the weed location and mark it in the system for future reference.
* **Pass/Fail Criteria**: The drone should reach the correct position within a predefined tolerance range (e.g., 1-2 meters).

**e) Test Case 5: Return to Base After Full Scan**

* **Objective**: Ensure the drone returns to the base after completing a full scan of the farm.
* **Preconditions**: Drone has completed scanning the farm area.
* **Test Steps**:
  1. Verify that the drone has scanned the entire farm area.
  2. Ensure the system recognizes that the farm area is fully scanned.
  3. Command the drone to return to its base.
  4. Verify that the drone successfully returns to the base without issues.
* **Expected Result**: The drone should autonomously return to the base after completing its task.
* **Pass/Fail Criteria**: The drone should return to the base within an acceptable time frame and without errors.

**Chapter 6**

**Limitations**

1. **Environmental Factors:**

* This aspect of the system's functionality would be affected by adverse weather and, possibly, changes in illumination including those that may make light a shadow or bright.

1. **Detection Precision:**

* The presence of false positive and negative cases regarding detection or classification of weeds challenges and, particularly when a similarity has to be between crop types and weed, that one might treat inaccurately.

1. **Hardware Limitations:**

* The high computational demands of the real-time weed detection, besides the limited battery life, limit operational efficiency, processing capabilities, and even field coverage during long operations.

1. **Data Dependency:**

* It is the quality and quantity of annotated data that train robust models. Poor-quality data or missing diverse annotations can limit effective generalization of the system about various scenarios.

1. **Cost:**

* The high initial investment and continuous maintenance costs for drones and software make the system less accessible for small-scale farmers, thus limiting its adoption in economically constrained regions.

1. **Limited Coverage Area:**

* Large agricultural fields may exceed the operational range of a single drone, requiring multiple drones or frequent battery recharges, which could lead to inefficiencies in time and resource management.

1. **System Integration:**

* Integration with existing farm management platforms or workflows can be difficult because of compatibility issues, requiring further customization or software adjustments.

1. **Real-Time Processing:**

* Video frames or action execution delays can lead to a delay in weed identification and treatment, thus reducing the responsiveness and effectiveness of the system.

1. **Herbicide Control:**

* Despite the precision spraying techniques, there is a possibility of herbicide wastage or unintended environmental contamination, which could undermine the sustainability goals of the system.

1. **Scalability:**

* Expanding the system to accommodate larger or more diverse agricultural areas adds complexity and cost, necessitating additional resources and enhanced infrastructure to maintain performance at scale.

**Chapter 7**

**Future Scope**

1. **Enhanced Detection Accuracy:**

* Advancements in computer vision algorithms and deep learning models offer the potential to significantly improve weed identification and classification, especially under challenging conditions. These improvements can enhance detection accuracy in low light, varying weather, and when crops or weeds overlap, enabling more reliable and consistent performance across diverse farming environments.

1. **Integration with IoT and Smart Farming:**

* By linking the weed detection system to Internet of Things (IoT) devices such as soil sensors and weather stations, farmers can access a wealth of real-time data. This integration offers comprehensive insights into environmental conditions, soil health, and crop growth, helping farmers make more informed decisions regarding weed management and overall farm operations, thus improving the efficiency of weed control.

1. **Support for Diverse Crops and Weeds:**

* Expanding the system’s dataset to include a greater variety of crops and weed species will increase its adaptability to diverse agricultural environments. This expansion would allow the technology to be used in a broader range of crops and regions, offering farmers more flexibility and precision when addressing different types of weed infestations.

1. **Real-Time Monitoring and Updates:**

* Incorporating 5G technology or edge computing into the system could significantly improve its ability to process and transmit data quickly, allowing real-time weed monitoring across large fields. This would enable farmers to receive timely updates on weed status, make quick decisions, and ensure efficient, continuous monitoring of their crops without delays.

1. **Cost Reduction:**

* Developing more affordable hardware solutions, such as drones with lower costs or energy-efficient models, could make the system accessible to small and medium-scale farmers. By reducing the overall cost of deployment and maintenance, the technology can become more economically feasible for a larger portion of the agricultural community, promoting broader adoption.

1. **Integration with Robotics:**

* Coupling weed detection and classification systems with autonomous robotics, such as smart tractors or robotic weed removal units, could automate the entire process of weed management. This automation would further reduce the need for manual labor, increasing operational efficiency and reducing costs, while providing farmers with a more comprehensive, all-in-one solution for weed control.

1. **Sustainable Practices:**

* Enhancing the system’s precision for herbicide application can help minimize chemical use, ensuring that herbicides are applied only where needed. This targeted spraying approach reduces waste and lowers the environmental impact of farming operations, aligning with sustainable agricultural practices and promoting environmental stewardship.

1. **Global Applicability:**

* By adapting the system to meet various geographical conditions and regulatory standards, it can be tailored to fit different regions across the globe. Whether in terms of crop varieties, local regulations, or climate conditions, making the system globally applicable would increase its market reach and offer a versatile solution for farmers worldwide.

1. **User-Friendly Interfaces:**

* Developing intuitive, user-friendly interfaces for mobile or web-based dashboards can significantly enhance the user experience for farmers. These interfaces would allow farmers to easily monitor field conditions, receive alerts about weed growth, and control operations efficiently, without the need for advanced technical knowledge, making the technology accessible to a broader user base.

1. **AI-Driven Recommendations:**

* Using predictive analytics and artificial intelligence (AI) can enhance the system by providing actionable recommendations for weed prevention based on historical data, environmental conditions, and predictive models. These AI-driven suggestions could help farmers implement proactive strategies, improving their overall efficiency and reducing the need for reactive weed control measures.

**Chapter 8**

**Conclusion**

The proposed system, 'Weed Detection and Control Using Computer Vision', has presented 85% accuracy with very promising discrimination results between weeds and crops. The system can efficiently and accurately identify different types of weed species under various environmental conditions by using state-of-the-art computer vision techniques such as deep learning-based object detection. A more robust and comprehensive dataset that incorporates weed and crop types has also maximized the accuracy of the detection system. This precision gives the system an edge not just as a reliable tool to the farmer but also cuts off a huge amount of required manual labor, thus aiding in improving the efficiency levels of weed management practices.

In addition, this will also lead to the system to recognize and classify weeds with greater precision, thus making herbicide application more targeted and effective and reducing the chemical use during agricultural processes. This also yields significant environmental benefits inasmuch as it helps the farmer adopt sustainable farming practice by minimizing the harmful impact of herbicides on soil and the ecosystem. Furthermore, the reduction in chemical usage leads to better crop yield and healthier farm ecosystems. While there is always room for further refinement and improvement in accuracy, especially in complex field conditions, the current performance highlights the system's considerable potential as an essential tool for modern agriculture, offering significant benefits both economically and environmentally.

**Chapter 9**

**Bibliography**

1. **Books:**

* Bishop, Christopher M. *Pattern Recognition and Machine Learning.* Springer, 2006.
* Goodfellow, Ian, et al. *Deep Learning.* MIT Press, 2016.

1. **Research Papers Reference:**
   1. A. Kamilaris and F. X. Prenafeta-Boldú, "Deep Learning in Agriculture: A Survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
   2. A. Milioto, et al., "Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging RGB-D Data," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 2343–2350, Oct. 2019.
2. **Additional References:**

* Roboflow, Inc. *Dataset Management and Model Training Platform.* Accessed on 1st November 2024, Available at: <https://roboflow.com>