ISSN 2277-2685
IJESR/June. 2025/ Vol-15/Issue-3s/638-642
M Spandana *et. al.*, / International Journal of Engineering & Science Research

Violence Detection

Dr P Sumalatha, M Spandana, M Sri Navya

[1] Associate Professor&Hod, Department Of CSE, Bhoj Reddy Engineering College For Women, India.

[2,3,4]B. Tech Students, Department Of CSE, Bhoj Reddy Engineering College For Women, India.

ABSTRACT

In today's digital era, the proliferation of video content on surveillance systems and social media platforms has raised serious concerns about public safety, especially in identifying and preventing violent activities. Manual monitoring of such video data is not only labor-intensive but also prone to human error and delayed response. To address this challenge, we propose an automated Violence Detection System using deep learning techniques that enhances surveillance capabilities by efficiently identifying violent behavior in video streams.

Our system leverages the power of Convolutional Neural Networks (CNN) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for temporal sequence analysis. The CNN model, specifically VGG16, is employed to extract high-level features from individual frames of a video, while LSTM processes the sequential data to understand motion dynamics and context over time. Together, this hybrid architecture enables accurate classification of video clips into violent or non-violent categories.

The backend is built using Python (Flask framework) and integrated with a user-friendly web interface that allows users to upload video files for analysis. The model provides outputs in the form of classification results along with a confidence score and inference time. Video data is preprocessed through a pipeline that includes frame extraction, resizing, noise reduction, normalization, and sequence grouping. To handle data imbalance issues, techniques like augmentation and synthetic sampling are applied.

This system not only automates violence detection but also supports scalability, real-time processing, and integration with existing surveillance infrastructures. By deploying it in public spaces, transport hubs, or educational institutions, the proposed model can significantly contribute to early detection and prevention of violent incidents, thereby improving public safety and emergency response.

2. REQUIREMENT ANALYSIS

Functional Requirement

1. User Interface Module

Functionality:

- Provides a web-based platform for users to interact with the system.
- Enables users to analyze multiple videos by restarting the process from the dashboard.

Key Features:

- Built using HTML, CSS, JavaScript (frontend) and Flask (backend).
- Simple, responsive, and accessible even for non-technical users.

2. Video Preprocessing Module

Functionality:

- Handles the transformation of uploaded video files into a format suitable for model input.
- Groups frames into temporal sequences that represent the progression of events.

Non-Functional Requirements

- **Usability**

- The system must have a simple, intuitive interface that can be used by individuals with minimal technical knowledge.

- **Reliability**

- It should deliver consistent and accurate results with minimal error, even when processing large video files.

- **Scalability**

- The system must support increasing volumes of users and video data without affecting performance.

- **Security**

- It must ensure secure handling of user data and uploaded videos, protecting against unauthorized access or data breaches.

- **Portability**

- The system should run on various platforms (Windows, macOS, Linux) without major code changes or compatibility issues..

Hardware Resources

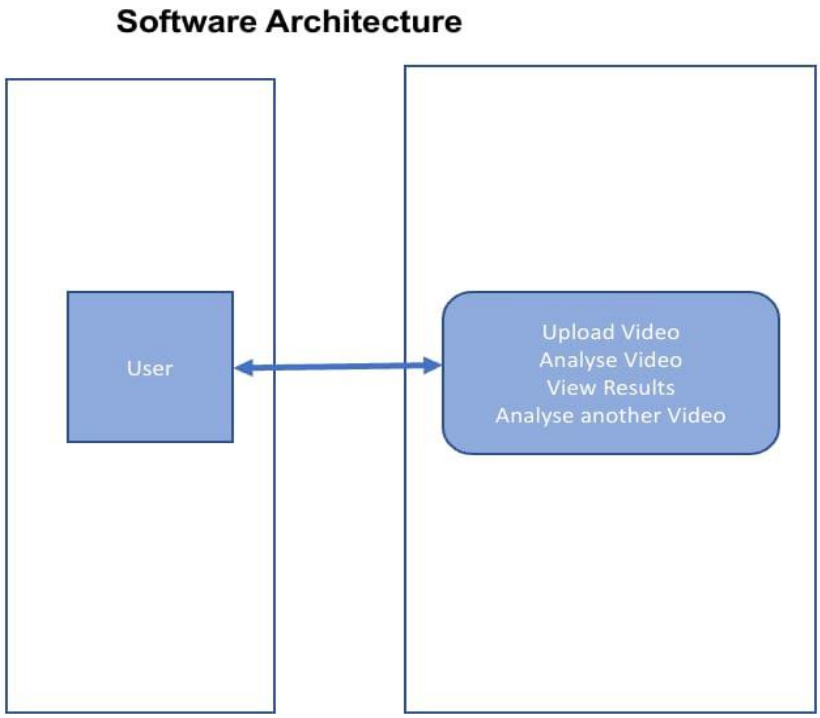
1. Processor : intel i3 and above
2. RAM : 4GB
3. Hard Disk : 500 GB

Software Resources

1. Operating System : Windows 11
2. Programming Language : Python 3.8.19
3. IDE : Anaconda
4. Framework : Flask
5. Database : MySQL

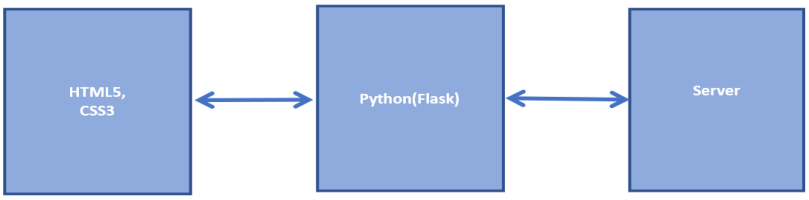
3. DESIGN

Software Architecture



3.1.1 Software architecture

Technical Architecture



3.1.2 Technical Architecture

4. IMPLEMENTATION

Here's a detailed breakdown of the technologies used in the architecture

The implementation of the Violence Detection Network for Video Surveillance required a wide range of technologies from multiple domains, including machine learning, computer vision, web development, and data handling. This section provides a comprehensive breakdown of each tool, library, and technology used, along with their specific role and justification within the project.

Programming Language: Python 3.8.19

Python was chosen as the primary language for both backend development and deep learning model implementation. It offers a highly readable syntax, a vast ecosystem of libraries, and seamless integration with machine learning frameworks.

Why Python?

- Strong support for AI and ML through libraries like TensorFlow, Keras, and PyTorch.
- Easily integrates with OpenCV for video processing and Flask for web development.
- High productivity and maintainability due to clean and concise syntax.

Use Cases in the Project:

- Writing video preprocessing scripts.
- Building and training the VGG16 and LSTM models.
- Managing backend logic and API endpoints in Flask.
- Handling data manipulation and performance evaluation tasks.

Web Framework: Flask

Flask is a lightweight, Python-based micro web framework that serves as the backend for the system. It facilitates communication between the frontend interface and the machine learning model hosted on the server.

Why Flask?

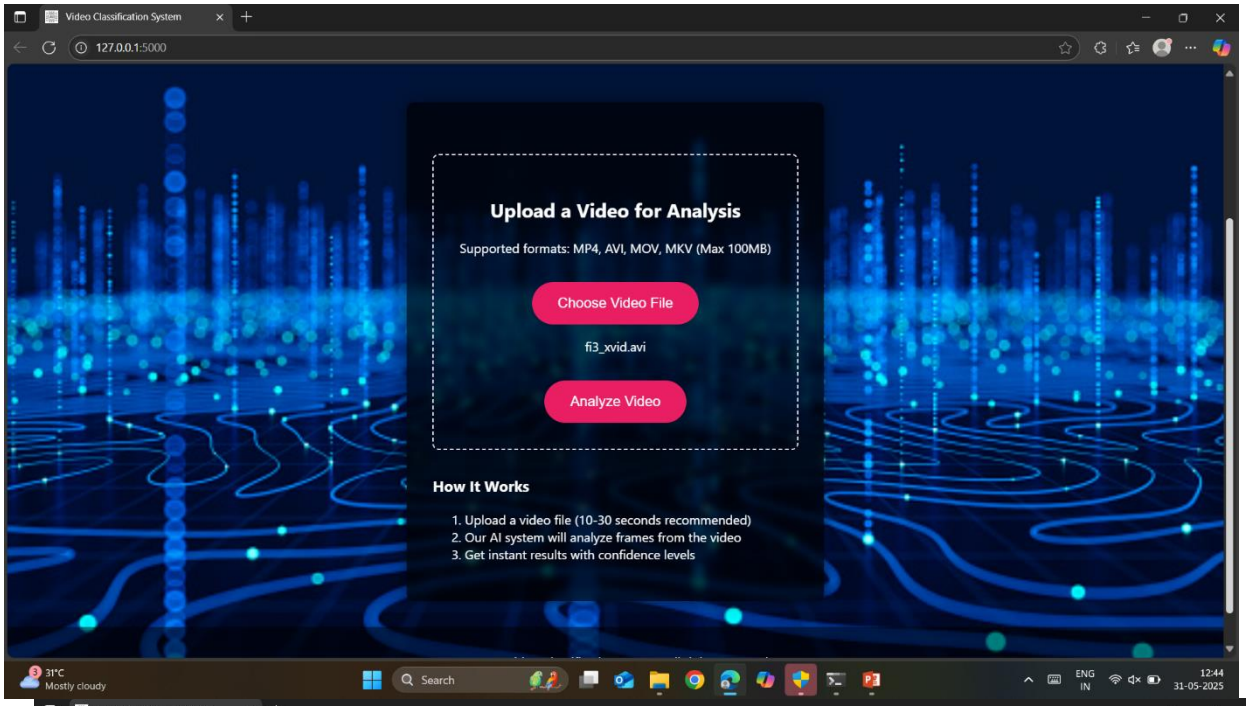
- Minimal and flexible framework, ideal for ML project deployment.
- Supports RESTful APIs, file handling, and integration with machine learning models.
- Allows quick prototyping and is easy to scale for larger applications.

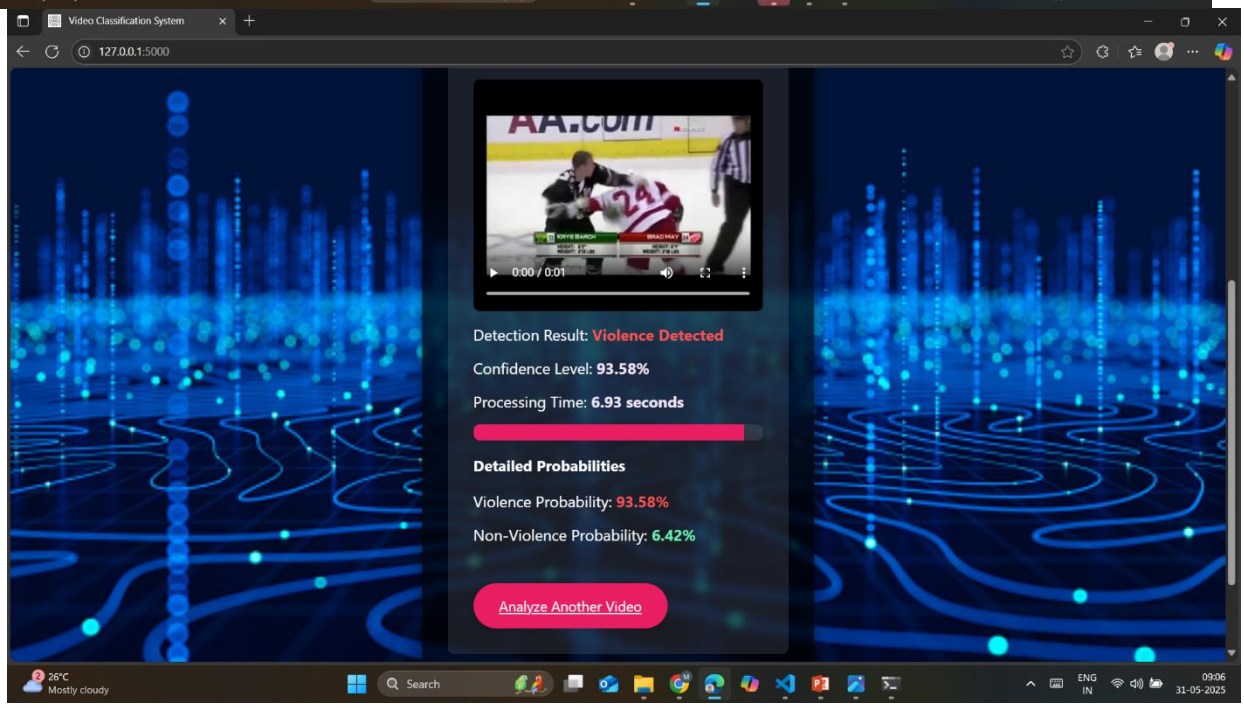
Frontend Technologies: HTML5, CSS3, JavaScript

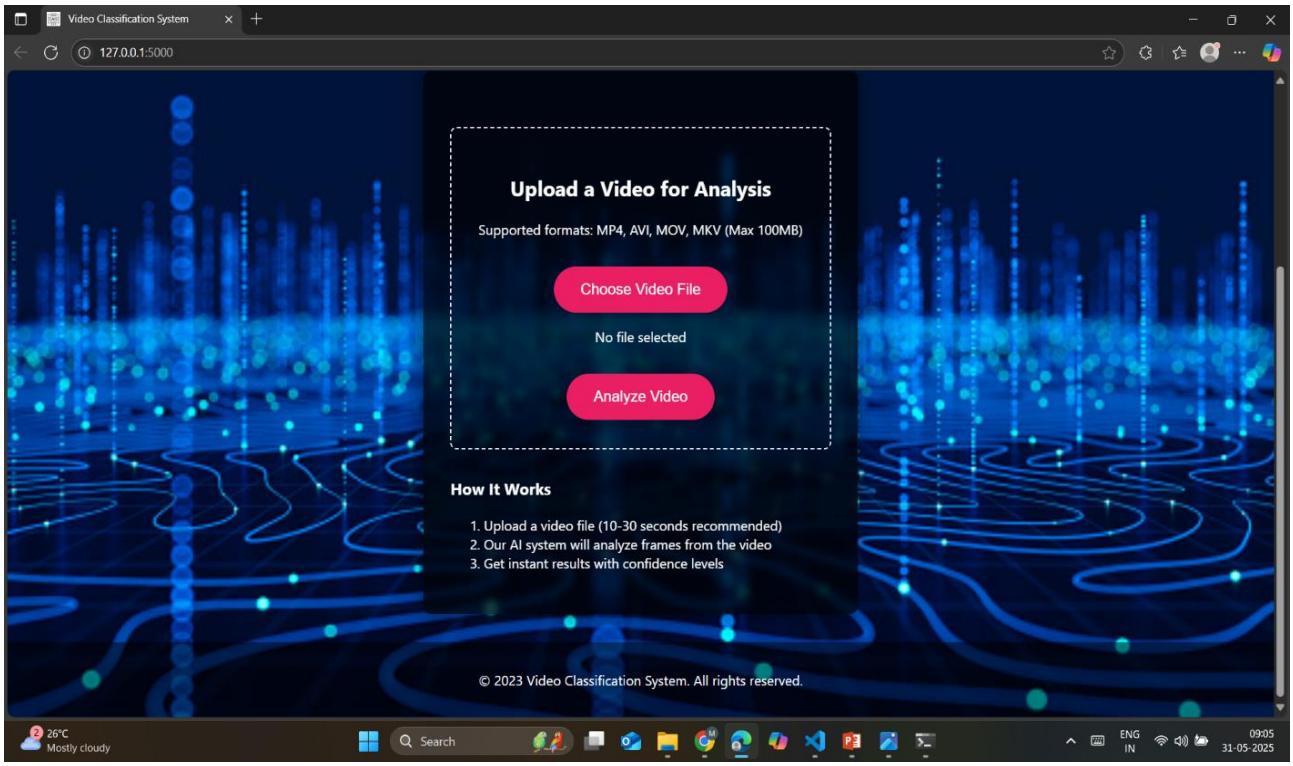
The **user interface** of the system was developed using standard web technologies, ensuring it is

accessible through any modern browser and requires
no additional installations.

5-SCREENSHOTS







REFERENCES

[1] G. Pathak, A. Kumar, S. Rawat, and S. Gupta, "Streamlining Video Analysis for Efficient Violence Detection," *arXiv*, Dec. 2024.

[2] R. Vijeikis, V. Raudonis, and G. Dervinis, "Efficient Violence Detection in Surveillance," *Sensors*, vol. 22, no. 6, Jun. 2022.

[3] M. Cheng, K. Cai, and M. Li, "RWF-2000: An Open Large Scale Video Database for Violence Detection," *arXiv*, Nov. 2019.

[4] S. Sudhakaran and O. Lanz, "Learning to Detect Violent Videos using Convolutional Long Short-Term Memory," *arXiv*, Sep. 2017.

[5] H. M. Bin Jahlan and L. A. Elrefaei, "Detecting Violence in Video Based on Deep Features Fusion Technique," *arXiv*, Apr. 2022.

[6] "IEEE Reference Style Guide: Numbering and ordering," *Scribbr*, 2023.