PREMIER UNIVERSITY CHATTOGRAM

DEPARTMENTOF **COMPUTER SCIENCE AND ENGINEERING**

PROJECT PROPOSAL

COURSE NAME	Artificial Intelligence Laboratory
COURSE CODE	CSE 3318
PROJECT NAME	AI-Based Real-Time Weapon and Gesture Detection System for Emergency Alerts
DATE OF SUBMISSION	03-12-2024

SUBMITTED TO

Avisheak Das

Lecturer

DEPARTMENT OF Computer Science and Engineering

SUBMITTED BY

NAME	Shadman Al Muksit, Rehenuma Tarin Tuhi, Priyam Sharma, Samira Sultana Mysha, Sariful Islam
ID	0222220005101003, 0222220005101005, 0222220005101018, 0222210005101022, 0222220005101023
SEMESTER	5th
BATCH	42
SESSION	Fall 2024
SECTION	A

Introduction:

In recent years, the need for enhanced security measures in public spaces has become increasingly apparent. The integration of artificial intelligence (AI) in security systems offers promising solutions for real-time threat detection and emergency response. This project proposes the development of an AI-based real-time weapon and gesture detection system designed to improve safety in public areas by providing immediate alerts during potential threats.

Objectives:

The main objectives of this project are:

- To develop a real-time detection system capable of identifying weapons using computer vision techniques.
- To implement gesture recognition for identifying distress signals or emergency gestures. - To create a user-friendly interface for security personnel to receive alerts and monitor detected threats.
- To evaluate the system's effectiveness in real-world scenarios through testing and validation.

Background and Motivation:

The rise in gun violence and public safety threats necessitates innovative approaches to enhance security. Traditional surveillance systems often lack real-time analysis capabilities, leading to delayed responses. This project aims to address these gaps by utilizing AI technologies, including deep learning and computer vision, to facilitate timely interventions in emergency situations.

Methodology:

System Design:

1. Data Collection:

- Gather a diverse dataset of images and videos for weapon detection and gesture recognition.
- Utilize publicly available datasets and generate synthetic data if necessary.

2. Model Development:

- Implement convolutional neural networks (CNNs) for weapon detection.
- Use recurrent neural networks (RNNs) or similar architectures for gesture recognition.
- Train models using transfer learning techniques to improve accuracy.

3. Real-Time Processing:

- Integrate the trained models into a real-time processing pipeline using OpenCV and TensorFlow or PyTorch.
- Optimize models for efficient inference on edge devices or cloud services.

System Implementation:

- Develop a user interface for security personnel to receive alerts and view detected threats.
- Implement alert mechanisms, such as SMS or app notifications, for emergency responses.
- Conduct extensive testing in various environments to assess system performance.

Development Tools:

- Programming Languages: Python (for AI model development).
- Libraries/Frameworks: OpenCV, TensorFlow, PyTorch, YOLOv5 for real-time detection. - Hardware:

  For training: Google Colab / Kaggle Notebooks. For deployment: USB camera, old PC/laptop

Workflow:

Data Collection:

- Curate datasets of weapon images and human gestures from public repositories and real-world recordings.

Model Training:

- Train detection models using transfer learning with existing architectures like YOLO or SSD.

Integration:

- Integrate models with real-time video processing pipelines.

Alert System:

- Implement a notification system using APIs (Twilio, Firebase, etc.).

Testing:

- Validate system performance in simulated and real-world scenarios.

Expected Outcomes:

- A fully functional AI-based weapon and gesture detection system that provides real-time alerts.
- Documentation of system performance, including accuracy, speed, and reliability.
- Recommendations for future improvements and potential applications in different security contexts.

Challenges and Risks:

Data Availability: Limited datasets for specific gestures and weapons may affect model performance.

False Positives/Negatives: Ensuring reliability in diverse lighting and environmental conditions.

Ethical Concerns: Privacy issues related to real-time surveillance and video processing.

Problem Statement:

Despite advancements in surveillance systems, traditional approaches lack real-time detection capabilities and proactive measures. Key limitations include:

Delayed Response:

Security personnel may fail to detect threats immediately.

Human Error:

Manual monitoring is prone to fatigue and oversight.

Inefficiency in Gesture Detection:

Existing systems often fail to detect non-verbal cues like aggressive gestures.

Timeline:

Phase	Duration
Data Collection	Flexible, depending on data availability and team effort
Model Development	Flexible, contingent on training progress and iteration
System Implementation	Flexible, based on ation challen ges integ r and team collaboration
Testing and Evaluation	Flexible, depending thoroughness of on testing and adjustments
Final Report Preparation	Flexible, based on the needed for time documentation and review

Budget:

Item	Estimated
Data Acquisition	Flexible, based on data sources and collection methods
Hardware (Cameras, etc.)	Variable, depending on and specifications quantities needed
Software Licenses	Adjustable, based on required and tools licenses
Miscellaneous	Uncertain, depending on unforeseen expenses
Total	Flexible, subject to adjustments based on project requirements

Conclusion:

The proposed AI-based real-time weapon and gesture detection system represents a significant advancement in public safety technology. By harnessing the power of AI, this project aims to provide timely alerts during emergencies, thereby enhancing the effectiveness of security measures. With proper execution, this system holds the potential to save lives and improve response times in critical situations.

References:

- [1] Zhang, X., et al. (2020). "Real-Time Object Detection for Public Safety Surveillance."
- [2] LeCun, Y., et al. (2015). "Deep Learning." Nature.
- [3] OpenCV Documentation: https://docs.opencv.org/
- [4] TensorFlow Documentation: https://www.tensorflow.org/