International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

INTRUDER DETECTION SYSTEM USING MACHINE LEARNING

Ms. T. S. Atre[1], VAISHNAVI NIKAM[2], VIDYA NIKAM[3] , NIKITA THORAT[4] , AMATYA TANDALE [5]

DEPARTMENT OF COMPUTER ENGINEERING GES'S R H SAPAT COE MANAGEMENT STUDIES AND RESEARCH KRISHI NAGAR, NASHIK

Abstract - The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisher face method. Facial image recognition Eigenface method is based on the reduction of face dimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project face detection system with face recognition is Image processing. The software requirements for this project is MATLAB software.

Keywords: *Digital Image Processing, Face Detection, Face Recognition, Motion Detection*

1.INTRODUCTION

The evolution of digital technologies has brought about significant advancements in security measures, with intruder detection systems (IDS) being at the forefront of these innovations. Intruder detection systems are designed to identify unauthorized access or suspicious activities within a secured environment, be it physical premises or digital networks. Traditional IDS often rely on predefined rules and signatures to detect threats, which can be limited in their ability to adapt to new and evolving threats.

Machine learning (ML) introduces a transformative approach to enhancing the capabilities of IDS by leveraging data-driven algorithms that can learn and adapt over time. Unlike conventional systems, machine learning-based IDS can analyze vast amounts of data, identify patterns, and make predictions with a higher degree of accuracy and efficiency. This adaptive learning ability is crucial in today's dynamic threat landscape where attackers constantly develop new techniques to bypass security measures.

By integrating machine learning into intruder detection systems, organizations can benefit from improved threat detection rates, reduced false positives, and the ability to detect previously unknown threats. Machine learning models can process various data types, including network traffic, user behavior, and environmental sensors, to provide a comprehensive security solution. Furthermore, these systems can be trained continuously with new data, enhancing their ability to recognize and mitigate sophisticated intrusion attempts.

The implementation of machine learning in IDS involves several steps, including data collection, feature extraction, model training, and continuous monitoring. Various machine learning algorithms, such as supervised learning, unsupervised learning, and reinforcement learning, can be applied depending on the specific requirements and nature of the data. Additionally, techniques such as anomaly detection, clustering, and classification play vital roles in identifying and responding to potential security threats.

2. LITERATURE SURVEY

Rishabh Paunikar, Shubham Thankare, Utharsh Abuse from Computer Engineering Bharati Vidyapeeth College of Engineering, Navi Mumbai. In April 2020 they made a complex project named "Action Recognition Using Surveillance System" the project was based on ideas in which the neural network with different datasets twice for computing an action. For object detection, the yoloV3 [3] will be used to detect humans and or any other objects and for activity recognition. The same neural network processing Yolo[2] weights will be used again which will be trained on a different dataset with human actions. They got the conclusion of the project demonstrating pattern matching, object detection, and action recognition.

Nandhini R, Duraimurugan N, S.P.Chokkalingam from International Journal of Engineering and Advanced Technology (IJEAT). In February 2019 they made a project named "Face Recognition Based Attendance System". It was on Automatic face recognition (AFR) technologies have made many improvements in the changing world. Smart Attendance using Real-Time Face Recognition. They

developed the "Fingerprint Based Recognition" later hours or before, the student needs to record the fingerprint on the configured device to ensure their attendance for the day. The problem with this approach is that during the lecture time it may distract the attention of the students then they implemented "Radio Frequency Identification" using a connection of RS232. 7

K.Govinda 1 K., Sai Krishna Prasad, Sai ram susheel SCSE, University, Vellore, India, SENSE, VIT University Vellore, India. They Created a nice project named "Intrusion Detection System for Smart Home using Laser Rays" In March 2014 whereas it is issued in vol. 2in IJSED (International Journal for Scientific Research & Development) it is based on the device that works based on the interaction between the sensor (which is LDR) and light source, preferably a LASER. When light is incident upon the LDR that is connected-the resistance would below, which directs a high input current through the base of the transistor, which in turn gives a low output which is accepted as an input into the buzzer. The conclusion of the program is most occasions the security system was usually occupied or organized by big insurance companies or specific security companies.

3. REQUIREMENT AND ANALYSIS

1. Objectives

- **Detect unauthorized access**: Identify when unauthorized entities access or attempt to access the system.
- **Real-time alerts**: Provide immediate notifications of detected intrusions.
- **Minimize false positives**: Ensure that legitimate activities are not incorrectly flagged.
- **Scalability**: Handle increasing amounts of data and expand across multiple environments.
- **Adaptability**: Evolve with new types of threats and intrusions.

2. Functional Requirements

- **Data Collection**:
  - Collect data from various sources (e.g., network traffic, system logs, sensor data).
  - Ensure continuous data flow for real-time monitoring.
- **Data Preprocessing**:
  - Clean and normalize data to ensure consistency.
  - Feature extraction to identify relevant attributes for detection.
- **Intrusion Detection**:
  - Implement machine learning models to classify normal and abnormal behavior.
  - Use techniques such as anomaly detection, supervised learning, or unsupervised learning.
- **Alert System**:
  - Generate real-time alerts upon detecting an intrusion.
  - Provide detailed reports including timestamp, type of intrusion, and affected areas.
- **User Interface**:
  - Develop a dashboard for monitoring system status and alerts.
  - Include visualization tools for data analysis and reporting.

3. Non-Functional Requirements

- **Performance**: Ensure the system can process data and generate alerts with minimal latency.
- **Scalability**: Design the system to accommodate growing data volumes and more data sources.
- **Security**: Protect the system against tampering and ensure the integrity of the detection process.
- **Reliability**: Ensure high availability and fault tolerance to avoid downtime.
- **Usability**: Create an intuitive interface for ease of use by security analysts.

4. System Architecture

- **Data Sources**: Network devices, servers, IoT devices, user behavior logs.
- **Data Ingestion**: Tools for streaming and batching data into the system.
- **Data Storage**: Databases and data lakes for storing raw and processed data.
- **Preprocessing Layer**: Modules for data cleaning, normalization, and feature extraction.
- **Detection Engine**: Machine learning models and algorithms for identifying intrusions.
- **Alerting Mechanism**: Real-time notification system integrated with email, SMS, or other communication tools.
- **User Interface**: Web-based dashboard for real-time monitoring and historical data analysis.

5. Machine Learning Models

- **Supervised Learning**: Models trained on labeled data to identify known intrusion patterns.
- **Unsupervised Learning**: Algorithms to detect anomalies in unlabeled data.

- **Hybrid Approaches**: Combining multiple techniques to improve detection accuracy.

6. Data Management

- **Data Collection**: Implement APIs and agents to collect data from diverse sources.
- **Data Storage**: Use scalable storage solutions like Hadoop, HDFS, or cloud storage.
- **Data Retention**: Define policies for data retention and archival based on regulatory requirements.

7. Evaluation Metrics

- **Accuracy**: Measure the correctness of intrusion detection.
- **Precision and Recall**: Evaluate the trade-off between false positives and false negatives.
- **F1 Score**: Balance precision and recall for overall performance.
- **Latency**: Time taken to detect and alert about an intrusion.

8. Implementation Plan

- **Phase 1**: Requirement gathering and feasibility study.
- **Phase 2**: System design and architecture setup.
- **Phase 3**: Data collection and preprocessing module development.
- **Phase 4**: Development of machine learning models.
- **Phase 5**: Integration of detection engine and alert system.
- **Phase 6**: User interface development and testing.
- **Phase 7**: System testing and validation.
- **Phase 8**: Deployment and monitoring.

9. Risk Management

- **Data Privacy**: Ensure compliance with data protection regulations.
- **Model Drift**: Regularly update models to cope with evolving threats.
- **System Overload**: Implement load balancing and scaling solutions.
- **False Positives/Negatives**: Continuously refine models to improve accuracy.

4. METHODOLOGIES/ALGORITHM DETAILS

Motion Detection: One commonly used algorithm for motion detection is the Background Subtraction method, which compares each video frame with a background model to identify moving objects. Other methods like Optical Flow or Frame Difference can also be utilized. Face Detection: The project can employ popular face detection algorithms such as Haar cascades or the more advanced methods like Convolutional Neural Networks (CNNs) or Histogram of Oriented Gradients (HOG) to detect human faces within the video frames. Face Recognition: For face recognition, the project can utilize algorithms like Eigenfaces, Fisher faces, or Local Binary Patterns (LBP) to extract facial features and compare them with the database of known individuals. Deep learning-based approaches such as Siamese Networks or Face Net can also be employed for more accurate and robust face recognition. Alarm Triggering: When a covered face is detected or an unrecognized face is identified, an alarm can be triggered using appropriate audio or visual indicators. This can be achieved through simple logic-based rules or machine learning techniques for decision-making.

Algorithm 1/Pseudo Code

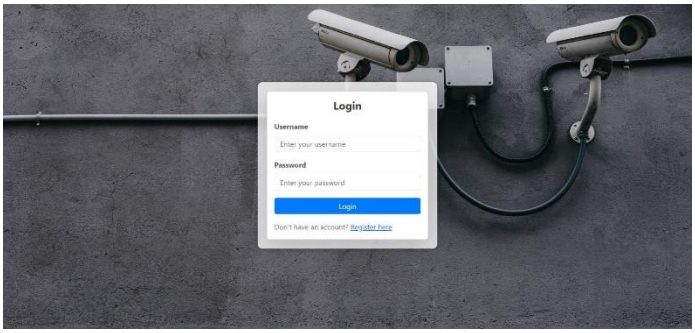
from flask import Flask, render_template, url_for, redirect, Response,request, after_this_request from tensorflow.keras.applications.mobilenet_v2 import preprocess_input from tensorflow.keras.preprocessing.image import img_to_array from tensorflow.keras.models import load_model import numpy as np import pygame import imutils import cv2 import os app = Flask(name) cap = None @app.route('/name', methods=['GET','POST']) def name(): return render_template('name.html') @app.route('/name_video',methods=['GET','POST']) def name_video(): return Response(name_detection(cap), mimetype='multipart/x-mixed-replace; boundary=frame') @app.route('/index',methods=['GET','POST']) def index(): @after_this_request def close_camera(response): release_camera() return response return render_template('index.html') @app.route('/',methods=['GET','POST']) def first(): return redirect(url_for('start')) @app.route('/mask', methods=['GET']) def mask(): return render_template('mask.html') @app.route('/mask_video', methods=['GET']) def mask_video(): return Response(mask_frames(cap), mimetype='multipart/x-mixed-replace; boundary=frame') @app.route('/motion', methods=['GET']) def motion(): return render_template('motion.html') @app.route('/motion_video',methods=['GET']) def motion_video(): return

Response(motion_detection(cap),mimetype='multipart/x-mixed-replace; boundary=frame') @app.route("/start",methods=["GET", "POST"]) def start(): return render_template("start.html") @app.before_request def before_request(): 31 init_camera() if name == ' main ': app.run(debug=True)

5.IMPLEMENTATION

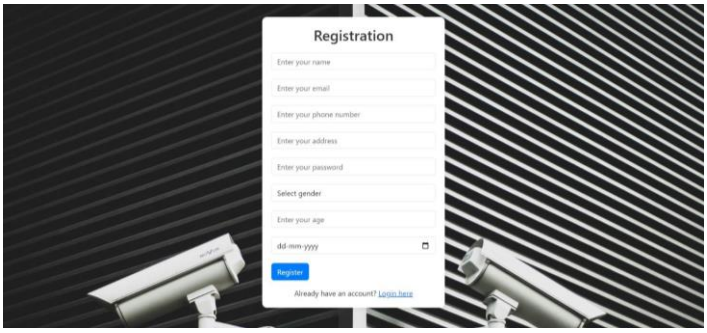
1. Login Page

The login page is the entry point for authorized users to access the Intruder Detection System. It ensures that only authenticated users can monitor and manage the system.



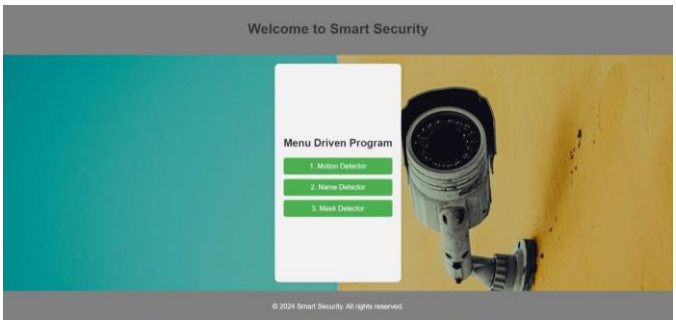
2. Registration Page

Username Field: A text input for the user to choose a unique username.Email Field: A text input for the user to enter their email address.Password Field: A password input for the user to set a password. Typically, a second password confirmation field is used to ensure the user types their password correctly.

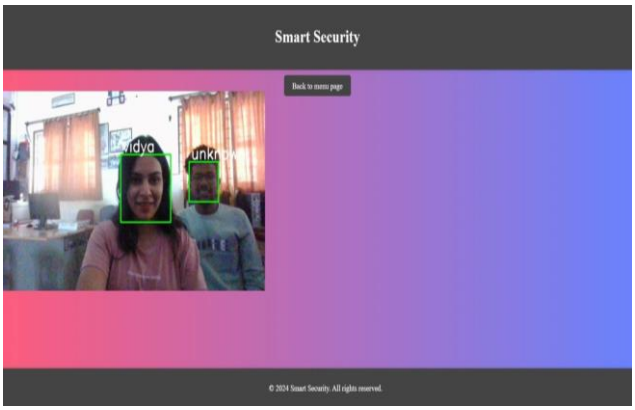


3. Dashboard

Navigation Menu: Links or buttons to navigate between different sections of the dashboard, such as Home, Settings, Alerts, and Reports.Live Feed/Video Monitoring: A real-time display of the areas under surveillance, showing video feeds from security cameras.

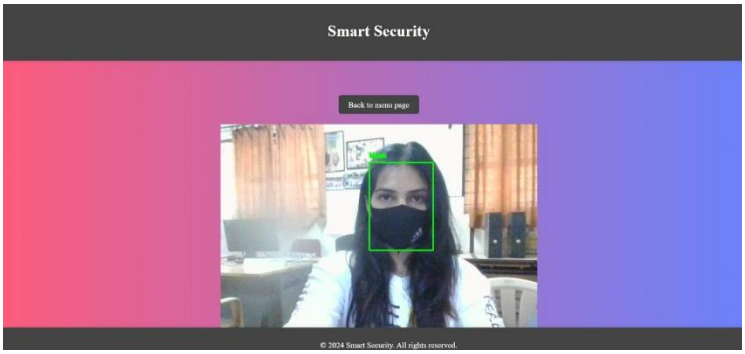


4. Output



5. Detection

5.1 Mask Detected



5.2 Mask Not Detected

CONCLUSION

We have developed an Intruder Detection System (IDS) using machine learning techniques, demonstrating its potential for enhancing security in various applications. The primary goal was to detect unauthorized access and anomalies within a system, which was achieved by leveraging the capabilities of supervised and unsupervised learning algorithms.

Our IDS model was trained and evaluated on a dataset comprising normal and intrusive behavior patterns. We implemented several machine learning algorithms, including decision trees, support vector machines (SVM), and neural networks, and conducted comprehensive evaluations to compare their performance. The results indicated that neural networks, particularly deep learning models, provided the highest accuracy in identifying intrusions, thanks to their ability to capture complex patterns in the data.

REFERENCE

[1] Real-Time Face Mask Detection using OpenCV and DeepLearning, Department of ECE, KoneruLakshmaiah Education Foundation, Andhra Pradesh, India.

[2] An Intelligent Motion Detection Using OpenCV. International Journal of Scientific Research in Science, Engineering and Technology Print ISSN: 2395-1990

[3] Y. You, S. Gong, C. Liu, "Adaptive moving object detection algorithm based on back ground subtraction and motion estimation", Int. J. Advancements in Computing Technology, vol. 5, no. 6, pp. 357-363, 2013 Conference, WWW2019,2019.

[4] 9M. Murshed, A. Ramirez, O. Chae, “Statistical Background Modeling: An Edge Segment Based Moving Object Detection Approach”, Proc. of IEEE International Conf. on Advanced Video and Signal Based Surveillance, pp. 300-305, 2010

[5] Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam, “Real-Time Object Detection with Yolo”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249– 8958, Volume-8, Issue-3S, February 2019.

[6] Marengoni and Stringhini. High Level Computer Vision using OpenCV. 2011. Universidade Presbiteriana Mackenzie.

[7] Design a face recognition system, The 15th International Conference on Machine Design and Production June 19 – 22, 2012, Pamukkale, Denizli, Turkey.