International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

ENSURING ACADEMIC INTEGRITY: SOULTION FOR ONLINE EXAM SECURITY USING DEEP LEARNING TECHNIQUES

Prof. R. A.[1], A.Sanjana[2],B.Sanjana[3],P.Sanjana[4],U.Sanjana[5],R.Sanjay[6]

[1] Manikandan Department of AIML School of Engineering Malla Reddy University
[2] 2111CS020477 School of Engineering Malla Reddy University
[3] 2111CS020478 School of Engineering Malla Reddy University
[4] 2111CS020479 School of Engineering Malla Reddy University
[5] 2111CS020480 School of Engineering Malla Reddy University
[6] 2111CS020480 School of Engineering Malla Reddy University

ABSTRACT:

The shift to remote learning has created new challenges in maintaining academic integrity during online exams. This innovative solution addresses these concerns by providing a secure, reliable, and scalable online exam experience. Leveraging advanced technologies such as facial recognition, visual tracking, and artificial intelligence AI-powered automated proctoring, this system detects and prevents cheating in real-time. The facial recognition feature verifies test-taker identity, while visual tracking continuously monitors the screen and surroundings to prevent unauthorized assistance. AI-powered algorithms analyze test-taker behavior, flagging suspicious activity and ensuring the integrity of the exam process. With its scalable architecture, this solution supports large-scale exam delivery, making it an ideal choice for educational institutions and organizations. By maintaining academic integrity, promoting fairness, and reducing cheating, this innovative solution helps to ensure the validity and reliabily of online exams. The proposed AI-powered proctoring solution addresses the pressing need for secure online exams in remote learning environments. By integrating advanced technologies such as facial recognition, visual tracking, and AI algorithms, the system ensures the integrity of the exam process, promotes fairness, and reduces the incidence of cheating. Its scalable architecture makes it an ideal choice for educational institutions of all sizes, helping to uphold the validity and reliability of online exams.

INTRODUCTION

The shift to remote learning has brought about unprecedented opportunities for flexibility and accessibility in education, enabling students to learn from virtually anywhere. However, this transformation has also exposed significant vulnerabilities in the examination process. Traditional methods of monitoring, designed for in-person settings, have struggled to adapt to remote environments, creating gaps in security and oversight. These challenges have led to increasing incidents of cheating, doubts about the credibility of assessments, and concerns about fairness among students. To address these pressing challenges, we propose an AI-powered proctoring solution that integrates advanced facial recognition algorithms, gaze-tracking techniques, and machine learning-based behavior analysis. The system uses computer vision (CV2) for accurate facial recognition and identity verification, visual tracking powered by computer vision to monitor screen and environmental activity, and AI-driven anomaly detection algorithms to flag suspicious behaviors in real-time. This comprehensive approach ensures a secure and reliable online exam environment by combining cutting- edge technologies with robust monitoring capabilities. Designed for educational institutions, certification bodies, and organizations of all sizes, this solution promotes fairness, upholds academic integrity, and provides a scalable framework for large-scale online exam delivery. For instance, a university conducting remote semester-end exams can use the system to verify student identities through facial recognition, monitor their screen activity with visual tracking, and analyze their behavior using AI algorithms. This ensures that exams are conducted securely and fairly, even with thousands of students participating from different locations. By meeting the growing demand for secure remote assessments, this solution supports the future of equitable and trustworthy education

LITERATURE REVIEW

1. Fastovets, D.V., Bogdanov, Y.I., Bantysh, B.I. and Lukichev, V.F., 2019, March. Machine learning methods in quantum computing theory. In *International Conference on Micro-and Nano-Electronics 2018* (Vol. 11022, pp. 752-761). SPIE.
2. Singh, J., Ali, F., Shah, B., Bhangu, K.S. and Kwak, D., 2022. Emotion quantification using variational quantum state fidelity estimation. *IEEE Access*, *10*, pp.115108-115119. Singh, J., Ali, F., Shah, B., Bhangu, K.S. and Kwak, D., 2022. Emotion quantification using variational quantum state fidelity estimation. *IEEE Access*, *10*, pp.115108-115119.
3. Ruskanda, F.Z., Abiwardani, M.R., Mulyawan, R., Syafalni, I. and Larasati, H.T., 2023. Quantum-enhanced support vector machine for sentiment classification. *IEEE Access*.

4. Liu, Y., Li, Q., Wang, B., Zhang, Y. and Song, D., 2023. A survey of quantum- cognitively inspired sentiment analysis models. *ACM Computing Surveys*, *56*(1), pp.1-37.

5. Ruskanda, F.Z., Abiwardani, M.R., Syafalni, I., Larasati, H.T. and Mulyawan, R., 2023. Simple Sentiment Analysis Ansatz for Sentiment Classification in Quantum Natural Language Processing. *IEEE Access*, *11*, pp.120612-120627.

PROBLEM STATEMENT

The transition to online education has transformed how academic institutions, certification bodies, and organizations conduct assessments. Online exams offer students unprecedented flexibility and accessibility, enabling them to take tests from virtually anywhere. However, this shift has also introduced significant challenges in maintaining academic integrity, as remote examinations are inherently more vulnerable to malpractice. Unlike traditional in-person exams, where invigilators can closely monitor students, online exams lack direct supervision, leading to increased risks of cheating, impersonation, and unfair advantages.

METHODOLOGY

The proposed system integrates advanced algorithms and frameworks for real-time face recognition, object detection, and voice activity monitoring. It is designed with modularity, scalability, and robustness in mind, targeting applications such as remote proctoring, home security, and corporate surveillance. Below, each aspect of the system is detailed:

Face Recognition Module:
This module is responsible for verifying user identity by comparing live webcam feeds with a pre-trained facial recognition model. It is crucial for ensuring that the individual being monitored matches the registered user.

Key Components

- *Data Collection*:
  - Users register by capturing multiple face images under varying conditions (e.g., lighting, expressions).
  - Images are preprocessed using Haar cascades or Dlib to ensure alignment and cropping.
  -
- *Model Training*:
  - The system employs the LBPH (Local Binary Patterns Histogram) algorithm for face recognition.
  - LBPH extracts texture features from images and trains a model to associate them with user IDs.
- *Real-Time Recognition*:
  - During monitoring, frames from the webcam are processed to detect and recognize faces.
  - Confidence thresholds ensure the system detects mismatches or unregistered users.

Object detection Module:
The object detection module ensures no unauthorized objects are present in the monitored environment. For example, during an exam, it verifies that only permitted items are present.

Key Components

- *Pre-Trained Model*:
  - The system leverages YOLO (You Only Look Once), a state-of-the-art real-time object detection model.
  - YOLO is pre-trained on the COCO dataset, which covers 80 object classes, including "person," "cell phone," and "laptop."
- *Detection Pipeline*:
  - Frames from the webcam are fed into the YOLO model to identify and localize objects.
  - Detected objects are cross-referenced with a whitelist of permitted items.
- *Violation Logging*:
  - Continuous detection of unauthorized objects triggers warnings and logs the anomaly.

Voice Activity detection Module:
This module identifies unauthorized speech or noise, ensuring compliance with silent environments during monitoring (e.g., exams or meetings).

Key Components

- *Feature Extraction*:
  - The system extracts MFCCs (Mel Frequency Cepstral Coefficients) or spectrograms from audio streams for analysis.
- *Speech Detection*:
  - Binary classification algorithms determine whether speech or background noise is present.
  - Advanced models like Wav2Vec can be integrated for speech-to-text transcription to detect specific phrases or unauthorized discussions.
- *Alerts*:
  - Continuous voice detection beyond a threshold (e.g., 10 seconds) triggers alerts and logs violations.

System Integrety:
The three modules are seamlessly integrated to provide real-time monitoring:

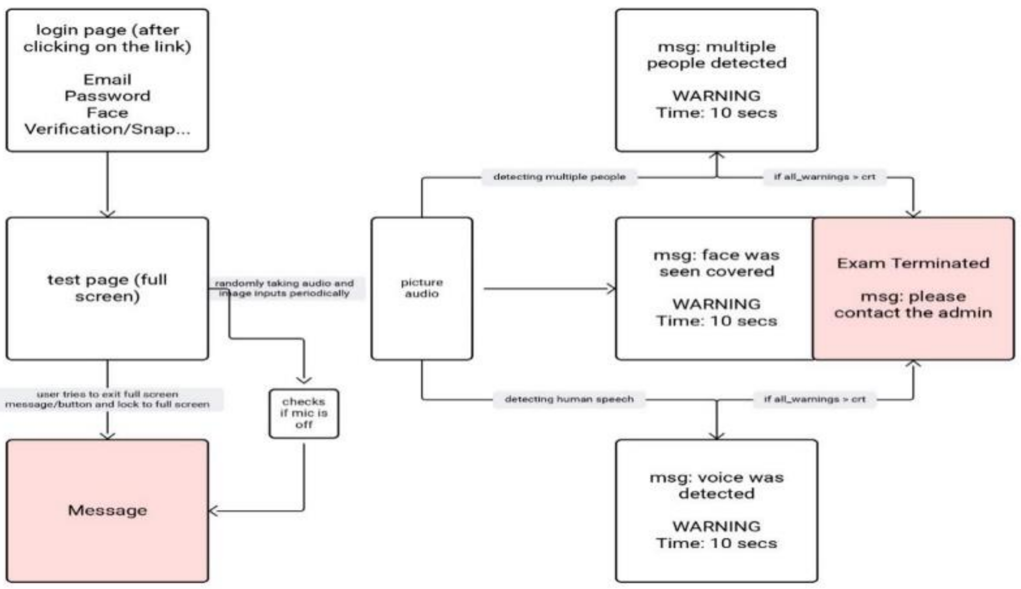
- *Input*: The system captures webcam video feeds and microphone audio streams.
- *Processing*:
  - Frames are simultaneously analyzed for faces, objects, and voice activity.
  - Multi-threading ensures efficient resource utilization and low latency.
- *Output*:
  - Alerts are displayed in the web interface, and violations are logged for future reference.

DATA ANALYSIS

The dataset is a crucial component of the application, as it provides the foundation for training and testing the machine learning models used for facial recognition, object detection, and voice recognition. The dataset requirements and usage are tailored to the specific tasks of user verification, anomaly detection, and continuous monitoring. Below is a detailed overview of the dataset requirements for each module of the application.
The facial recognition module requires a dataset containing multiple images of each user. These images are captured during the registration process to account for variations in lighting, facial expressions, and angles.The dataset is stored in a secure directory (user_data) on the server, and the LBPH algorithm uses these images to train a model for each user. This ensures that the face recognizer can accurately differentiate between authorized and unauthorized users.

MODAL DEVELOPMENT: TESTING AND VALIDATION

MODEL SELECTION AND ARCHITECTURE:


login page (after clicking on the link)
Email
Password
Face
Verification/Snap...
test page (full screen)
randomly taking audio and image inputs periodically
user tries to exit full screen message/button and lock to full screen
checks if mic is off
Message
picture audio
detecting multiple people
if all_warnings > crt
msg: multiple people detected
WARNING
Time: 10 secs
msg: face was seen covered
WARNING
Time: 10 secs
Exam Terminated
msg: please contact the admin
detecting human speech
if all_warnings > crt
msg: voice was detected
WARNING
Time: 10 secs

The program employs a combination of methods and algorithms from computer vision, machine learning, and voice processing to achieve its objectives of face recognition, object detection, and continuous monitoring. Below is a detailed explanation of the key methods and algorithms used:

1. Face Recognition:

The face recognition functionality is critical for user registration, verification, and monitoring.

Algorithm: Local Binary Patterns Histogram (LBPH)

- ***Purpose:***
  LBPH is used to train a user-specific model during registration and recognize the user's face during verification and monitoring.
- ***Process:***
  1. ***Feature Extraction:***
     - Divide the input image into smaller grids.
     - Compute the Local Binary Pattern (LBP) for each grid by comparing each pixel to its neighbors.
     - ***C***onvert the binary patterns into decimal values to form histograms.
  2. ***Model Training:***
     - Aggregate the histograms from all grids to create a feature vector.
     - ***T***rain a face recognition model based on these feature vectors.
  3. ***Face Recognition:***
     - ***D***uring recognition, compare the histogram of the input face with stored histograms to find the best match.
- ***Advantages:***
  - ◆ Effective for handling varying lighting conditions.
  - ◆ Simple and computationally efficient.
- ***Implementation:***

OpenCV’s cv2.face.LBPHFaceRecognizer_create() is used for training and prediction.

2. Object Detection:

Object detection ensures unauthorized objects are identified in the monitoring process.

Algorithm: YOLO (You Only Look Once)

- *Purpose*:

  YOLO is used to detect and classify objects in real-time video frames.

- *Steps*:

  1. *Input Image Splitting*:
     - Divide the input frame into a grid.
  2. *Bounding Box Prediction*:
     - Each grid cell predicts bounding boxes for potential objects along with confidence scores.
  3. *Class Prediction*:
     - Classify detected objects into predefined categories (e.g., person, bag, laptop).
  4. *Non-Max Suppression*:
     - Remove overlapping boxes to retain only the most confident predictions.
- *Advantages*:

Real-time detection with high speed and accuracy.Simultaneous detection of multiple objects.

- *Implementation*:

Pre-trained YOLO models (weights and configuration files) are used with OpenCV for detection.

3. **Voice Detection:**

Continuous voice activity detection is implemented to monitor unauthorized speech during the monitoring phase.

Method: Energy-Based Voice Activity Detection

- *Purpose*:

Detect voice activity by analyzing the energy levels in audio signals.

- *Steps*:
  1. *Audio Signal Processing*:
     - Capture audio input through a microphone.
     - Compute the Short-Time Energy (STE)of the signal over small windows.
  2. *Thresholding*:
     - Compare STE values to a predefined threshold to determine if speech is presen.
  3. *Continuous Monitoring*:
     - Track speech activity over time and log warnings if speech persists beyond the defined threshold duration.

***Advantages*:**

- Simple and effective for detecting voice in quiet environments.
- Low computational complexity.

***Implementation*:**

- Python libraries like PyAudio or custom signal processing code are used for capturing and analyzing audio.

4. **Multi-face Detection:**

Detects and monitors the number of faces in a video frame to identify anomalies (e.g., multiple faces in a single-user environment).

Algorithm: Harr Cascade

- *Purpose*:

Detect faces in video frames during registration, verification, and monitoring.

- *Steps*:
  1. *Feature Extraction*:
     - Haar-like features (e.g., edge, line, rectangle patterns) are extracted from the input frame.
  2. *Cascade Classifier*:
     - A pre-trained classifier evaluates regions of the frame to identify faces.
  3. *Bounding Box Detection*:
     - Draw bounding boxes around detected faces.
- *Advantages*:

Light weight and effective for real-time face detection.

- *Implementation*:

OpenCV's Haar Cascade Classifier (haarcascade_frontalface_default.xml) is use.

5. **Error Handling Monitoring System:**

A custom monitoring logic integrates outputs from the above algorithms to ensure robust functionality.

Method: Time-Based Thresholding

- *Purpose*:

Detect and respond to anomalies (e.g., face mismatch, multiple faces, face obstruction) over time.

- *Steps*:
  1. Track the duration of detected anomalies using timestamps.
  2. Log warnings if anomalies persist beyond a predefined threshold (e.g., 10 seconds).
  3. Reset timers when anomalies resolve.
- *Advantages*:

  - Avoids false positives caused by transient conditions.
  - Improves system reliability by focusing on sustained anomalies.

6. **Real-Time Processing:**

The system integrates threading to enable concurrent execution of tasks such as video capturing, face/object detection, and updating the client interface.

Method: Multi-Threading

- *Purpose*:

Ensure smooth real-time performance by offloading intensive tasks (e.g., monitoring) to separate threads.

- *Implementation*:

  - Python's threading module is used to run monitoring tasks in the background without blocking the main server.

7. **Logging and Notifications:**

The system includes logging and real-time notifications to ensure transparency and user awareness.

Method: Logging and API Update

- Logs warnings (e.g., face mismatch, voice detection) to text files for debugging and analysis.
- Provides real-time monitoring updates to the client through RESTful APIs.

TEST AND VALIDATE

No.	Modules	Accuracy
1	Face Recognition	0.9833
2	Object Detection	0.6700
3	Multi-Face Recognition	0.5700
4	Voice Detection	0.7100

I. MODEL IMPLEMENTATION

5.1.1 Setup and Implementation

- Import required libraries (Flask, cv2, custom modules, etc.).
- Create a Flask app.
- Define a directory (USER_DATA_DIR) to store user-related data.
- Initialize global variables:
  - warnings: To store monitoring warnings.
  - monitoring_status: To track the completion status and messages of monitoring.

5.1.2 Helper Functions

Initialize Face Recognizer:

- Try to create an LBPH face recognizer.
- Raise an error if OpenCV's face module is not available.

5.1.3 Continues Monitoring Logic

- *Input:* User ID.
- *Setup:*
  - Initialize webcam (cv2.VideoCapture).
  - Load the face recognizer model and YOLO object detection model.
  - Define thresholds and monitoring criteria.
- *Monitoring Loop:*
  - Capture a frame from the webcam.
  - Convert it to grayscale and detect faces.
  - Detect objects using the YOLO model.
  - Validate the following in real-time:
    - *Face Mismatch:* Check if detected face matches the registered user.
    - *Multiple Faces:* Check for more than one face.
    - *Face Obstruction:* Check if no face is detected.
    - *Unauthorized Objects:* Detect objects other than people.
    - *Voice Activity:* Monitor for continuous voice detection.
  - Log warnings if anomalies persist for a threshold duration (10 seconds).
  - Exit the loop if a critical warning is logged.
- *Cleanup:*
  - Release the webcam and close OpenCV windows.
  - Update monitoring_status with the results.

5.1.4 Flask Routes

Home Page:

- *Route:* /
- Render the home page template (home.html).

User Registration:

- *Route:* /register
- *GET Request:* Render the registration form.
- *POST Request:*
  - Capture user images using the capture_img module.
  - Train the face recognition model using the train module.
  - Redirect to the home page after successful registration.

User Verification:

- *Route:* /verify
- *GET Request:* Render the verification form.
- *POST Request:*
    - Verify the user's face using the verify module.
    - If successful, start monitoring in a separate thread and redirect to the test page.
    - If verification fails, display an access-denied message.

Test Page:

- *Route:* /test/<user_id>
- Render the test page with the user ID.

Monitoring Status:

- *Route:* /monitor_status
- Return the current monitoring status and messages as JSON.

Result Page:

- *Route:* /result/<message>
- Render the result page with a message indicating the outcome of monitoring.

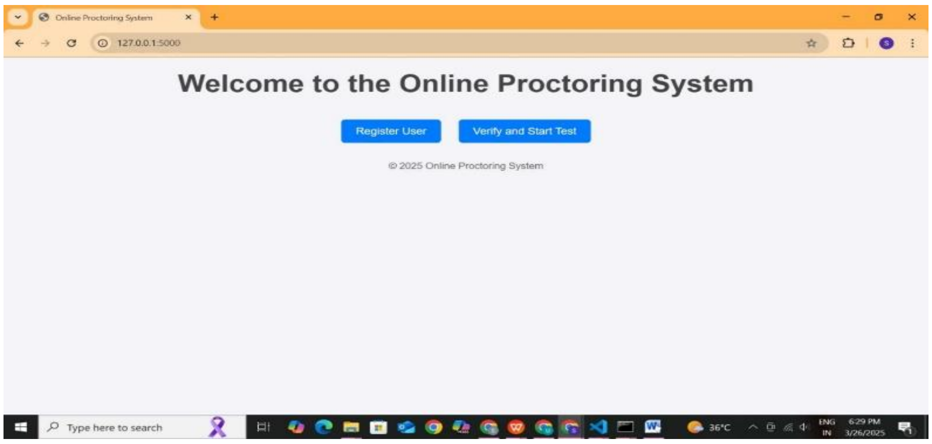
5.1.5 Main Application End Points

- Run the Flask app in production mode (debug=False).

5.1.6 Simplified flow diagram

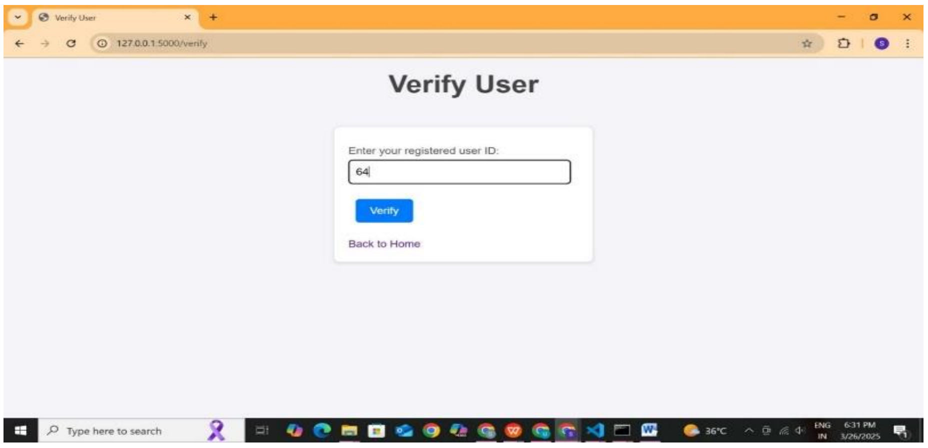
1. ***Home Page***
- User selects an action: Register, Verify, or Monitor.
2. ***Registration***
- Capture images → Train model → Save data.
3. ***Verification***
- Match user ID → If successful, start monitoring.
4. ***Monitoring***
- Continuously validate face, objects, and voice.
- Log warnings and display results.
5. ***Result***
- Display any anomalies or completion message.

RESULTS


Welcome to the Online Proctoring System

Register User Verify and Start Test

© 2025 Online Proctoring System

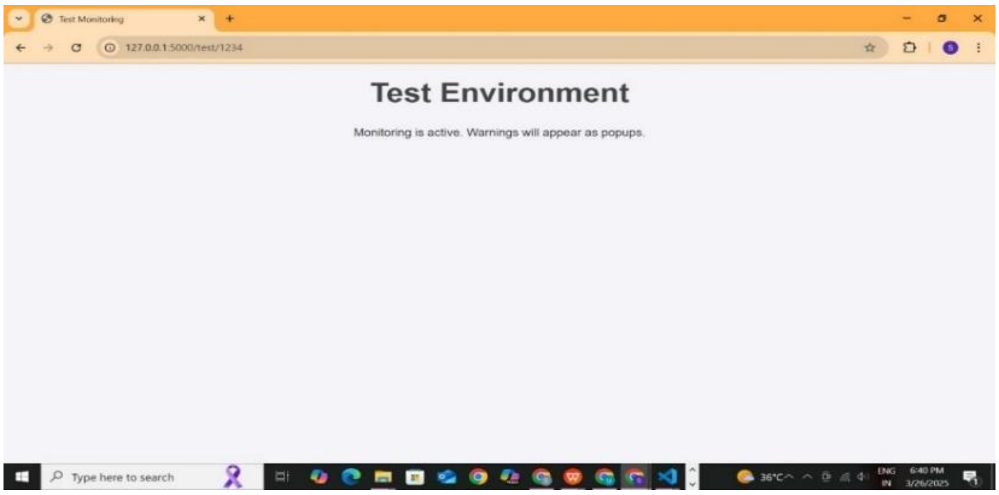

Verify User

Enter your registered user ID:

64

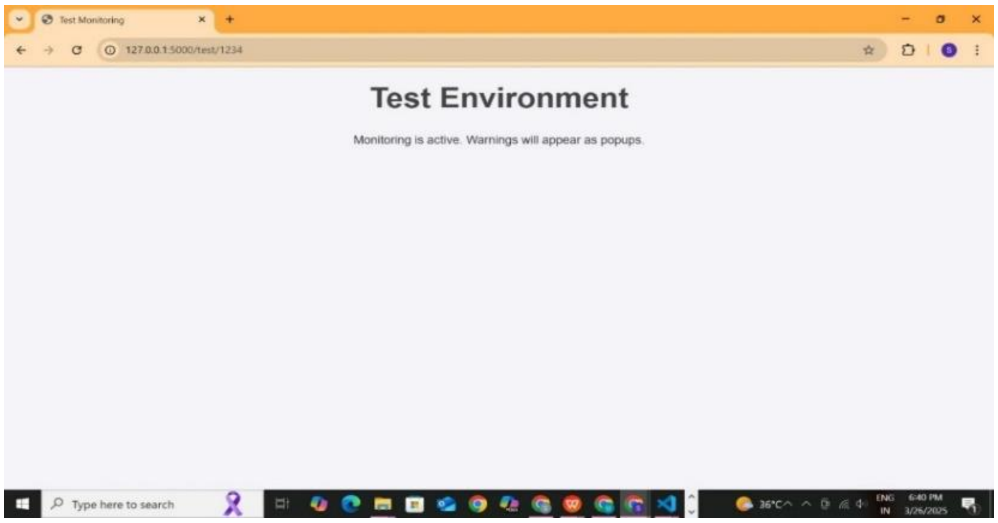
Verify

Back to Home



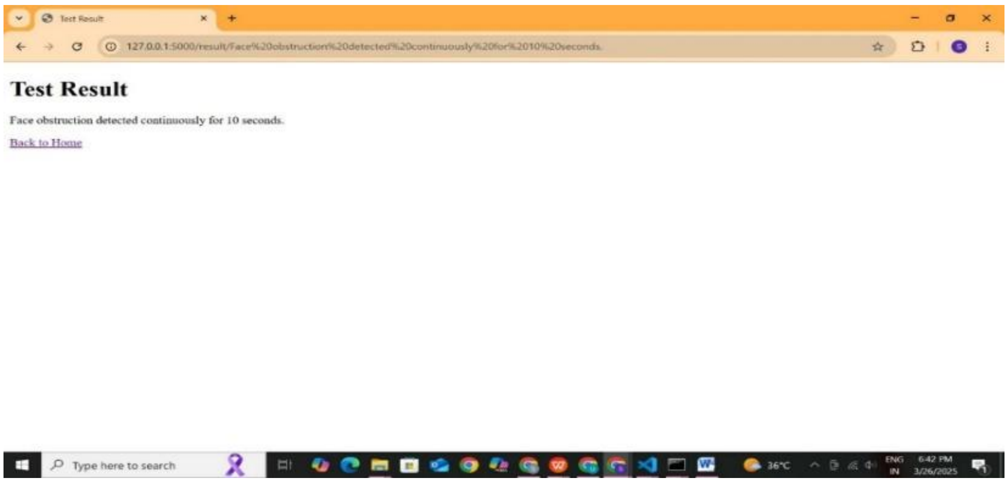
Test Environment

Monitoring is active. Warnings will appear as popups.



Test Environment

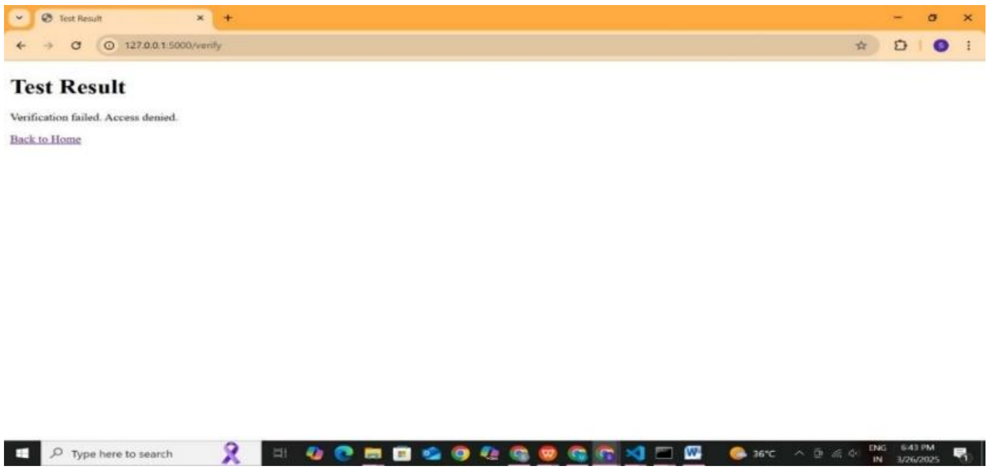
Monitoring is active. Warnings will appear as popups.


Test Result
127.0.0.1:5000/result/Face%20obstruction%20detected%20continuously%20for%2010%20seconds.
Type here to search
36°C
ENG IN
6:42 PM 3/26/2025

Test Result

Face obstruction detected continuously for 10 seconds.

Back to Home



Test Result

Verification failed. Access denied.

Back to Home

CONCLUSION

The developed program is a comprehensive system that integrates face recognition, object detection, and voice activity monitoring into a robust, modular, and scalable framework. By leveraging advanced algorithms such as LBPH for face recognition, YOLO for real-time object detection, and energy-based methods for voice activity detection, the system ensures accurate, efficient, and real-time monitoring for security and surveillance purposes.

From registration and verification to continuous monitoring, the program offers a seamless experience, ensuring user convenience and system security.The use of multi-threading and efficient algorithms ensures low-latency performance, making it suitable for real-time applications.The system effectively monitors multiple faces, face mismatches, unauthorized objects, and voice activity to detect anomalies, enhancing its reliability.Comprehensive error handling and logging mechanisms provide transparency, facilitate debugging, and ensure the system's robustness under different conditions.The modular design enables easy addition of new features, such as integration with databases or advanced analytics, ensuring long-term adaptability.

FUTURE ENHANCEMENT

The system's modular architecture allows for easy scalability and feature enhancements. Potential future improvements include:.

Integrating the system with cloud storage can provide a centralized logging and analysis solution. This would enable the storage of large amounts of data in a scalable and secure manner. The cloud storage can be used to store logs, metrics, and other data generated by the system, allowing for real-time monitoring and analysis. This can be particularly useful for identifying trends, detecting anomalies, and debugging issues. Furthermore, cloud storage provides automatic backups, disaster recovery, and collaboration features, making it an attractive solution for centralized logging and analysis.

Deploying the system on edge devices can significantly enhance its portability. Edge devices, such as smartphones, tablets, and single-board computers, can run the system in a standalone mode, without the need for a centralized server or cloud connection. This makes it ideal for use cases where internet connectivity is limited or unreliable, such as in remote areas, disaster zones, or industrial environments. By deploying the system on edge devices, users can take advantage of its features and capabilities, even in situations where traditional computing infrastructure is not available.

Utilizing advanced deep learning models can further enhance the accuracy and robustness of the system. Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown remarkable performance in various tasks, including image and speech recognition, natural language processing, and time series forecasting. By integrating these models into the system, users can expect even more accurate results, improved handling of complex data, and increased resilience to noise and outliers. Additionally, advanced deep learning models can enable the system to learn from large datasets, adapt to changing conditions, and make predictions with greater confidence.

REFERENCES

1. Fastovets, D.V., Bogdanov, Y.I., Bantysh, B.I. and Lukichev, V.F., 2019, March. Machine learning methods in quantum computing theory. In *International Conference on Micro-and Nano-Electronics 2018* (Vol. 11022, pp. 752-761). SPIE.
2. Singh, J., Ali, F., Shah, B., Bhangu, K.S. and Kwak, D., 2022. Emotion quantification using variational quantum state fidelity estimation. *IEEE Access*, *10*, pp.115108-115119. Singh, J., Ali, F., Shah, B., Bhangu, K.S. and Kwak, D., 2022. Emotion quantification using variational quantum state fidelity estimation. *IEEE Access*, *10*, pp.115108-115119.
3. Ruskanda, F.Z., Abiwardani, M.R., Mulyawan, R., Syafalni, I. and Larasati, H.T., 2023. Quantum-enhanced support vector machine for sentiment classification. *IEEE Access*.
4. Liu, Y., Li, Q., Wang, B., Zhang, Y. and Song, D., 2023. A survey of quantum- cognitively inspired sentiment analysis models. *ACM Computing Surveys*, *56*(1), pp.1-37.
5. Ruskanda, F.Z., Abiwardani, M.R., Syafalni, I., Larasati, H.T. and Mulyawan, R., 2023. Simple Sentiment Analysis Ansatz for Sentiment Classification in Quantum Natural Language Processing. *IEEE Access*, *11*, pp.120612-120627.
6. Baronia, D., 2023. Hybrid Quantum-Classical Neural Networks for Text Classification. *Authorea Preprints*.
7. Martinez, V. and Leroy-Meline, G., 2022. A multiclass Q-NLP sentiment analysis experiment using DisCoCat. *arXiv preprint arXiv:2209.03152*.
8. Zhang, Y., Song, D., Zhang, P., Li, X. and Wang, P., 2019. A quantum-inspired sentiment representation model for twitter sentiment analysis. *Applied Intelligence*, *49*, pp.3093-3108.
9. Chu, Z., Wang, X., Jin, M., Zhang, N., Gao, Q. and Shao, L., 2024. An Effective Strategy for Sentiment Analysis Based on Complex-Valued Embedding and Quantum Long Short-Term Memory Neural Network. *Axioms*, *13*(3), p.20