

GSO-CNN Face Recognition System

Abhishek Walke 1, Dhanshri Cholke2, Ankita Bhor 3, Swaleha Sayyed 4

1, Professor, Dept. of Computer Engineering, Vidya Niketan College of Engineering, Bota ,
2,3,4, B.E. Computer Engineering, Dept. of Computer Engineering, Vidya Niketan College of Engineering,
Bota.

Abstract -

Facial recognition systems are widely used in modern applications such as security systems, smart attendance, access control, and surveillance monitoring. However, many traditional systems face difficulties in recognizing multiple faces accurately in real-time environments due to variations in lighting, face angles, facial expressions, and image quality. To overcome these challenges, this project presents a real-time multiple face recognition system using Convolutional Neural Networks (CNN) integrated with Gravitational Search Optimization (GSO) for optimized feature selection and improved recognition performance.

The proposed system is developed using Python, OpenCV, OpenVINO, and Flask framework. The system captures facial data through a live camera feed, detects and recognizes multiple faces simultaneously, and compares them with stored face encodings in the database. CNN-based models are used for face detection and feature extraction, while GSO is applied to optimize important facial features from the extracted embeddings to improve recognition accuracy and reduce unnecessary feature complexity.

The system also provides a user-friendly web interface for adding new faces, managing the face database, and performing real-time face recognition. Experimental results show that the proposed system performs efficiently in real-time conditions and provides accurate recognition for multiple individuals appearing simultaneously in the camera feed. The developed system can be effectively used in smart surveillance, attendance management, and security-based applications.

Keywords: Facial Recognition, Multiple Face Recognition, Convolutional Neural Network (CNN), Gravitational Search Optimization (GSO), OpenVINO, OpenCV, Real-Time Recognition, Feature Optimization, Flask Framework, Computer Vision, Deep Learning, Face Detection.

1. INTRODUCTION

Facial recognition has become one of the most widely used applications in the field of computer vision and artificial intelligence. It is commonly used in security systems, smart attendance, access control, surveillance systems, and user authentication applications. With the increasing demand for automation and intelligent monitoring systems, real-time face recognition has gained significant importance in recent years.

Traditional facial recognition systems often face challenges in real-world environments due to changes in lighting conditions, facial expressions, camera angles, image quality, and partial occlusion. In many existing systems, recognition accuracy decreases when multiple

persons appear simultaneously in the camera feed. These limitations create the need for a more reliable and optimized facial recognition system capable of handling real-time multiple face recognition efficiently.

In this project, a real-time multiple face recognition system based on Convolutional Neural Networks (CNN) and Gravitational Search Optimization (GSO) is proposed. The system uses OpenVINO pre-trained models for face detection, landmark detection, and face re-identification. OpenCV is used for image processing and real-time video streaming, while Flask is used to develop a user-friendly web interface for system interaction.

The proposed system captures face images through a live camera feed and stores them in a face database. During recognition, the system detects multiple faces from the video stream, extracts facial feature embeddings using CNN models, and compares them with stored encodings in the database. The Gravitational Search Optimization (GSO) algorithm is used to optimize the extracted feature vectors by selecting important facial features and reducing unnecessary feature complexity. This helps improve recognition accuracy and overall system performance.

The developed system provides functionalities such as adding new faces, managing the face database, and performing real-time multiple face recognition through a web interface. The main objective of this project is to build an efficient, accurate, and practical facial recognition system that can work reliably in real-time environments for multiple users simultaneously.

2. LITERATURE SURVEY

Research on facial recognition systems has increased rapidly in recent years due to the growing demand for security, surveillance, attendance management, and smart authentication systems. Earlier systems mainly focused on basic face detection techniques, while recent approaches use deep learning and optimization algorithms to improve recognition accuracy in real-time environments. However, many systems still struggle with multiple face recognition, lighting variations, and real-time processing efficiency.

Al-Abboodi et al. [1] proposed a facial recognition system based on Convolutional Neural Networks (CNN) and Galactic Swarm Optimization (GSO). The system used GSO for feature optimization and improved recognition accuracy compared to traditional CNN models. Although the system showed good performance, it mainly focused on static image datasets and required further improvement for real-time multi-person recognition.

Khan et al. [2] developed a face recognition method using a multi-level 3D-GAN colorization approach. Their work improved image quality and facial feature extraction under low-quality image conditions. However, the computational complexity of the model was high, making real-time implementation difficult on standard systems.

Nguyen et al. [3] introduced a hybrid optimization method combining Galactic Swarm Optimization and Whale Optimization algorithms for global search problems. The proposed approach improved optimization performance and convergence speed, but the system required high computational resources for large-scale real-time applications.

Kim et al. [4] designed a real-time surveillance system capable of recognizing criminal faces from video streams. The system achieved good recognition performance in controlled environments, but accuracy decreased under varying lighting conditions and partial face occlusion.

Rahim et al. [5] implemented a CNN-based anomaly detection and face recognition system for smart home security applications. The proposed model provided reliable face recognition results, but the system performance depended heavily on dataset quality and camera positioning.

Chen et al. [6] proposed a feature extraction technique using Histogram of Oriented Gradients (HOG) with dimensionality reduction methods for signal and pattern recognition tasks. The method reduced unnecessary features and improved classification performance, but processing speed remained a challenge for live streaming applications.

Most of the existing systems focus mainly on single-face recognition or controlled datasets and do not provide an efficient solution for real-time multiple face recognition with optimized feature selection. Many approaches also require high-end hardware or cloud-based processing for better performance.

None of the reviewed works put together a complete offline pipeline—book upload, chapter split, topic embedding, explanation, test, score, and dashboard—in one lightweight package. The gap is clear: a modular, local-first system that handles real textbooks without internet or heavy hardware.

The proposed system addresses these limitations by combining CNN-based facial recognition with Gravitational Search Optimization (GSO) for feature optimization. The system is capable of detecting and recognizing multiple faces simultaneously in real-time using OpenVINO, OpenCV, and Flask framework while maintaining efficient performance and improved recognition accuracy.

3. SYSTEM MODEL

The proposed system is designed to perform real-time multiple face recognition using Convolutional Neural Networks (CNN) and Gravitational Search Optimization (GSO). The system is developed using Python, OpenCV, OpenVINO, and Flask framework. It provides a complete workflow for face detection, face data collection, database management, and real-time recognition through a web-based interface.

The system begins by capturing live video frames from a webcam or camera feed. The captured frames are processed using the OpenVINO face detection model to identify multiple faces present in the frame. Once the faces are detected, the system extracts facial feature embeddings using the face re-identification CNN model. These extracted features are then optimized using the Gravitational Search Optimization (GSO) algorithm to select the most important facial features and reduce unnecessary feature complexity.

The optimized feature vectors are compared with previously stored facial encodings available in the face database. If a match is found within the defined threshold value, the system displays the recognized person's name on the live video feed. If no match is found, the system labels the face as unknown. The proposed model supports simultaneous recognition of multiple persons appearing in the camera frame.

The system also includes a Flask-based web interface that allows users to add new faces, capture face images, manage the face database, and perform real-time recognition testing. During face registration, multiple face images of a person are captured and stored in the database to improve recognition reliability under different facial conditions and angles.

The overall workflow of the proposed system is divided into the following stages:

- **Live Video Capture**
- **Multiple Face Detection**
- **Facial Feature Extraction using CNN**
- **Feature Optimization using GSO**
- **Face Matching and Recognition**
- **Real-Time Output Display**
- **Face Database Management**

The proposed system improves recognition performance by combining deep learning-based feature extraction with optimization techniques. The integration of GSO helps reduce redundant facial features and improves recognition accuracy during real-time processing. The system can be effectively used in smart surveillance systems, attendance monitoring, security authentication, and access control applications.

4. IMPLEMENTATION:

The implementation of the proposed facial recognition system is carried out using Python, OpenCV, OpenVINO, Flask, and CNN-based face recognition models. The system is designed to perform real-time multiple face detection and recognition through a simple web interface. Different modules are connected together to handle face registration, database management, feature extraction, optimization, and live recognition.

The complete implementation process of the system is explained below.

A) Home Interface

The system starts with a Flask-based home interface that acts as the main dashboard of the application. From this page, the user can access different modules

such as adding new faces, viewing the face database, and testing real-time recognition. The interface is designed to keep the workflow simple and easy to use.

Technologies Used: Flask , HTML/CSS , Bootstrap , Python



Fig 1: Home Page

B) Face Registration Module

The face registration module is used to add new users to the system database. In this process, the user enters the person's name, after which the camera feed starts automatically. The system then captures multiple face images of the person from different angles and expressions.

Capturing multiple samples helps improve recognition performance during real-time testing. All captured images are stored in separate folders inside the face database.

Technologies Used: OpenCV, Python , Flask



Fig 2: Add Faces Module

C) Face Capture Process

The face capture process is responsible for collecting facial images through the live webcam feed. OpenCV is used for accessing the camera, while the OpenVINO face detection model identifies the face region from each frame.

After detection, the face area is cropped and saved automatically at fixed time intervals. The system captures around 50 images for each registered user to create a proper facial dataset for recognition.

Technologies Used: OpenCV, OpenVINO, CNN Face Detection Model

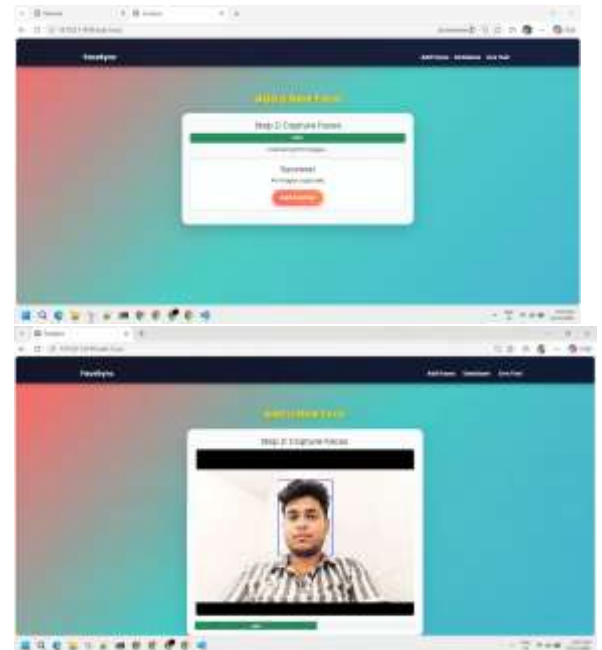


Fig 3: Face Capture and Dataset Generation

D) Face Database Management

The captured facial images are stored inside the face gallery database. Each person has a separate folder containing all captured face samples.

The database module allows users to view stored images, manage datasets, and remove unwanted records when needed. This helps maintain a clean and organized face dataset for recognition tasks.

Technologies Used: Flask, File Handling, OpenCV



Fig 4: Face Database Module

E) Real-Time Multiple Face Recognition

The real-time recognition module is the main part of the system. During testing, the camera feed continuously scans for faces present in the frame. The system can detect and recognize multiple persons simultaneously from the live video stream.

The OpenVINO CNN model extracts facial feature embeddings from detected faces. These feature vectors are then optimized using Gravitational Search Optimization (GSO), which helps remove unnecessary features and improves matching accuracy.

The optimized features are compared with stored face encodings available in the database using cosine distance matching. If a match is found, the recognized

person's name is displayed on the screen; otherwise, the face is marked as unknown.

Technologies Used: OpenVINO, CNN Models, OpenCV, Gravitational Search Optimization (GSO), Flask Streaming



Fig 5: Face Recognition

F) System Workflow

The overall workflow of the proposed system is divided into the following steps:

- User Registration
- Face Image Capture
- Face Dataset Storage
- Multiple Face Detection
- CNN-Based Feature Extraction
- GSO Feature Optimization
- Face Matching and Recognition
- Real-Time Output Display

The combination of CNN and GSO helps improve recognition accuracy, feature optimization, and real-time system performance. The developed system works efficiently for recognizing multiple persons appearing simultaneously in the camera feed.

5. RESULTS AND DISCUSSION

The proposed facial recognition system was tested in real-time using a webcam feed under different lighting conditions and face positions. The system successfully detected and recognized multiple faces simultaneously from the live camera stream. The implementation was carried out using OpenVINO models, OpenCV, Flask framework, and Gravitational Search Optimization (GSO) for feature optimization.

During testing, the system was able to capture facial images, generate the face database, and perform recognition with good accuracy. Multiple face recognition worked properly even when more than one person appeared together in the camera frame. The Flask-based interface also provided smooth navigation between different modules such as face registration, database management, and live testing.

The CNN-based feature extraction model generated facial embeddings for detected faces, while GSO optimized the extracted feature vectors by selecting important features and reducing unnecessary feature complexity. This helped improve recognition performance and reduced mismatching during real-time processing.

The system showed stable performance in normal indoor environments with proper lighting conditions. Recognition accuracy was better for registered users having sufficient face samples in the database. The use of multiple face samples during registration helped improve recognition reliability for different face angles and facial expressions.

The real-time recognition module continuously processed live video frames and displayed recognized names directly on the screen. The system was also able to identify unknown persons who were not present in the database.

The proposed system provides an efficient and practical solution for applications such as smart attendance systems, surveillance monitoring, access control, and security-based authentication systems. The integration of GSO with CNN models helped improve overall system performance and recognition accuracy compared to normal feature matching approaches.

6. ADVANTAGES AND LIMITATIONS

I. Advantages

- The system supports real-time multiple face recognition using live camera feed.
- CNN-based feature extraction helps improve face recognition accuracy.
- Gravitational Search Optimization (GSO) reduces unnecessary features and improves matching performance.
- The Flask-based web interface makes the system simple and user-friendly.
- Multiple face samples are captured during registration, which improves recognition under different facial conditions and angles.
- The system can be used for smart attendance, surveillance, access control, and security applications.
- OpenVINO optimization helps improve processing speed during real-time recognition.

II. Limitations

- Recognition accuracy may decrease under very low lighting conditions.
- The system performance depends on camera quality and face visibility.
- Large face databases may slightly increase recognition time.
- Extreme face angles or heavy face occlusion can affect recognition performance.
- The current system mainly supports single camera input at a time.

6. CONCLUSION

In this project, a real-time multiple face recognition system using CNN and Gravitational Search Optimization (GSO) was successfully developed. The system is capable of detecting and recognizing multiple persons simultaneously from a live camera feed using OpenVINO, OpenCV, Flask, and Python.

The use of CNN models improved facial feature extraction, while GSO helped optimize important facial features for better recognition accuracy and performance. The

system also provides a simple web interface for face registration, database management, and real-time testing.

The developed system can be effectively used in applications such as smart attendance, surveillance, and security systems. Future improvements can include multi-camera support and cloud-based database integration.

9. REFERENCES

- [1] R. H. R. Al-Abboodi and A. A. Al-Ani, "A Novel Technique for Facial Recognition Based on the GSO-CNN Deep Learning Algorithm," *Journal of Electrical and Computer Engineering*, vol. 2024, pp. 1–21, 2024.
- [2] Z. Khan, A. I. Umar, S. H. Shirazi, M. Shahzad, and M. Assam, "Face Recognition via Multi-Level 3D-GAN Colorization," *IEEE Access*, vol. 10, pp. 12504–12516, 2022.
- [3] B. M. Nguyen, T. Tran, T. Nguyen, and G. Nguyen, "Hybridization of Galactic Swarm and Evolution Whale Optimization for Global Search Problems," *IEEE Access*, vol. 8, pp. 126861–126872, 2020.
- [4] H. B. Kim, N. Choi, H. J. Kwon, and H. Kim, "Surveillance System for Real-Time High-Precision Recognition of Criminal Faces from Wild Videos," *IEEE Access*, vol. 11, pp. 71524–71535, 2023.
- [5] A. Rahim, Y. Zhong, T. Ahmad, S. Ahmad, P. Pławiak, and M. Hammad, "Enhancing Smart Home Security: Anomaly Detection and Face Recognition in Smart Home IoT Devices Using Logit-Boosted CNN Models," *Sensors*, vol. 23, no. 15, pp. 6979, 2023.
- [6] K. Chen, S. Chen, S. Zhang, and H. Zhao, "Automatic Modulation Recognition of Radar Signals Based on Histogram of Oriented Gradient via Improved Principal Component Analysis," *Signal, Image and Video Processing*, vol. 17, no. 6, pp. 3053–3061, 2023.