RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Project Report On

Missing Person Identification System

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By
Edwin M S (U2103082)
Gautham Sunilkumar(U2103091)
Gloriya Antony (U2103095)
Janis Reji(U2103106))

Under the guidance of

Ms. Sherine Sebastian

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
November 2024

CERTIFICATE

This is to certify that the project report entitled ***Missing Person Identification System*** *is a bonafide record of the work done by* ***Edwin M S (U2103082), Gautham Sunilkumar (U2103091), Gloriya Antony (U2103095), Janis Reji (U2103106)****, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2024-2025.*

Ms. Sherine Sebastian
Project Guide
Assistant Professor
Dept. of CSE
RSET

Ms.Anu Maria Joykutty
Project Coordinator
Assistant Professor
Dept. of CSE
RSET

Dr. Preetha K.G.
Professor & HOD
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **Dr Preetha K G**, Professor, Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our project, "Missing Person Identification System".

We are highly indebted to our project coordinator, **Ms. Anu Maria Joykutty**, Assistant Professor, Dept of CSE, for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Sherine Sebastian**, Assistant Professor, Department of Computer Science and Engineering, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Edwin M S
Gautham Sunilkumar
Gloriya Antony
Janis Reji

Abstract

The project introduces an advanced Missing Person Identification System that leverages cutting-edge face recognition technology to efficiently identify and locate missing individuals by comparing their photographs with archived CCTV footage. The system encompasses several key modules: a secure photo upload interface, a high-frequency video processing pipeline, and an image preprocessing unit that applies blur reduction techniques to optimize facial clarity. Using state-of-the-art deep learning models, such as Convolutional Neural Networks (CNNs), YOLOv8, and Transformers, the system accurately detects, extracts, and generates unique face embeddings from video frames for effective face matching.

In comparison to existing systems, this solution offers several critical advantages. First, its high-frequency frame extraction ensures no potential sightings are overlooked, even in fast-moving footage. Second, the system incorporates sophisticated image preprocessing and feature extraction to improve recognition accuracy under challenging conditions, such as low lighting and varying video resolutions. Additionally, performance optimization techniques minimize processing time, making the system scalable across large datasets and multiple camera feeds. Filtering tools with timestamps and data streamline the review process for law enforcement, accelerating response times. Overall, this project advances the state-of-the-art in missing person identification by enhancing matching accuracy, processing efficiency, and adaptability, thereby delivering a robust and scalable solution that significantly improves upon current identification methods.

Contents

List of Abbreviations

AI: Artificial Intelligence

CCTV:Closed-Circuit Television

CNN:Convolutional Neural Network

DNN:Deep Neural Network

EMG:Electromyography

FACS:Facial Action Coding System

GAN:Generative Adversarial Network

GMM:Gaussian Mixture Model

KNN:K-Nearest Neighbor

LDA:Linear Discriminant Analysis

MPIS:Missing Person Identification System

PCA:Principal Component Analysis

RRB:Reversible Residual Block

SVM:Support Vector Machine

VGG:Visual Geometry Group

YCbCr:Luminance-Chrominance Color Space

YOLO:You Only Look Once

List of Figures

List of Tables

Chapter 1

Introduction

Modern environments like campuses, cities, and huge public spaces are usually complex and large and derive unique challenges to public safety in case of missing persons. Most old-fashioned methods for locating missing individuals involve repeated manual surveillance and intervention which is slow, resource intensive, and always at risk with human error and have sparked appeal for fully automated solutions that can employ advanced technology to achieve greater accuracy and speed in search operations.

The system identifies the missing persons and integrates facial recognition technology to accomplish such identification over large areas expanse with the allowability of CCTV surroundings. All kinds of image processing techniques and their machine learning and deep learning techniques will be automated and used to analyze the CCTV footage so that missing persons can be detected quickly and accurately automatically by comparing their facial captures with live or recorded video streams.

The system intends to minimize the use of manual observation, allowing security personnel to focus on priority activities, in addition to providing a broader area coverage for surveillance. The automated identification simplified the identification process, thus greatly reducing the level of human error contributing to efficiency and responsiveness in security operations. The MPIS is a paradigm shift for public safety through its monitoring and rapid response mechanisms in cases when people get reported as missing.

1.1 Background

The identification and recovery of missing persons is still a major concern for law enforcement and public safety agencies around the world. Missing person's recovery methods are slow, ranging from manual CCTV review, public alerts, and extensive manual searches: all processes of locating individuals are sluggish because they often end up with each action taking a while and disallowing time management. They may even bring about improper response time, which will further reduce the chances of successful recovery, especially in having high traffic and densely populated areas. With the onset of more dense urban spaces and the crowds growing in the network of surveillance systems, there is mounting pressure to devise more effective automated approaches for rapid and effective identification of missing persons.

Recently, the emergence of artificial intelligence (AI) and computer vision especially around facial recognition has raised enthusiasm on automating person identification in nivella datasets such as those provided by CCTV systems. Facial recognition technology is proficiently detecting, analyzing, and matching facial components of an individual by computers. Therefore, scanning and identifying people in a network of camera feeds become an intelligent system. This emerging technology is increasingly being adopted for security and law enforcement applications meant at enhanced surveillance, faster investigations, and more efficient resource use. Despite such developments, there are many limitations such as poor image quality, variable illumination, and speed of processing in such systems, which inherently compromise accuracy and reliability of identification under real-life conditions.

Missing Person Identification System addresses all these problems by combining facial recognition with advanced, capable video processing to create a robust, scalable and efficient solution. It leverages multiple AI-driven modules for automating the processes of detection, extraction and matching of facial features recorded CCTV footage, thereby reducing human dependency on surveillance. Automatic high-frequency frame extraction, image preprocessing for clarity enhancement and feature extraction through CNNs, deep learning models, are some of the key technological components in this .

The MPIS potential is improving public safety by increasing both speed and accuracy in missing person identifications and therefore will assist law enforcement in a quicker response. Besides, the scalable architecture of the system makes it usable in a variety of environments-from large campuses and transit hubs to smart city infrastructures-where extensive CCTV coverage is available. It has brought about new standards in proactive public safety and missing person recovery-from human error and rapid processing of video data.

1.2 Problem Definition

The major concern discussed is that conventional ways of searching missing persons take time and also show a great deal of inefficiency along with being highly manual in their dependence. This often implies that the identification is done either very late or not properly. This project is targeted at developing an advanced automated Missing Person Identification System, which uses AI-powered facial recognition and video processing in the detection and identification of missing persons. This would enable faster and more reliable means of helping law enforcement agencies and society safety.

1.3 Scope and Motivation

The Missing Person Identification System is intended to be used in high-density public and private environments that experience much foot traffic and where an individual can easily get lost, like campuses, transit hubs, and urban places. This will reduce the alarm advance read-through of the integration with massive networks of CCTV for monitoring and analysis of video feeds and immediate notification when a possible match is detected with an image of a missing person. The MPIS system consists of secure photo upload, video processing, image preprocessing, and an advanced facial matching system through AI algorithms for accuracy and reliability. In addition to this, it was designed to be scalable to so that it could be used at both smaller facilities and with larger infrastructures having multiple camera feeds. Thus, it brings an all-encompassing automated solution for the need for a proactive as well as effective tool for missing person enquiry effective for various organizations and law enforcement agencies.

The project is really motivated by the gravest need for quick and simple solutions for cases of missing persons, where all seconds really matter. Most search operations now rely on manual monitoring and mostly suffer from human error, resulting in poor response rates and ineffective recovery. Given the growing advancements in artificial intelligence and computer vision, there exists a golden opportunity to blend these technologies and design an efficient system that minimizes delays while maximizing accuracy in recognition of the missing person. This is a project that is fundamentally committed to improving public safety and the optimization of resource allocation by equipping law enforcement with a tool that amplifies the robustness of CCTV monitoring. In the end, the MPIS will contribute significantly towards community safety with respect to the time and effort required to locate missing individuals by way of providing a technology-based solution to an important social issue.

1.4 Objectives

- **Create a secure and user-friendly platform** for uploading photographs, through which registered users upload of high-quality photographs of missing persons.
- **Develop a high-frequency pipeline for video processing** that extracts frames from archival CCTVs optimally for identification of lost persons from such footage.
- **Improvement of image quality** is done by means of various preprocessing techniques, such as denoising, deblurring, exposure normalizing, and aligning facial features, to enhance the recognition accuracy of the faces.
- **Develop and evolve deep learning algorithms** like CNN and Transformer for advanced detection of faces and feature extraction, leading to the accurate creation of facial embeddings for identification.
- **Set up face matching algorithms** that are accurate in their measure of similarity scores and threshold analysis such that detected faces can be matched to uploaded images without high levels of false positive results.
- **System Performance Optimization for Scalability and Efficiency** across Multiple Camera Streams and Alerts with Minimum Processing Delays for Timely Intervention by Law Enforcement.
- **Integrate Advanced Face Detection and Recognition** Utilize YOLOv8 (face.pt) for face detection and DeepFace for feature extraction and recognition.

1.5 Challenges

The major issues in the Missing Person Identification System (MPIS) are high accuracy in face recognition irrespective of the quality, illumination, and angles of capture with respect to the CCTV footage. Secondly the actual system should be capable of live processing since it is going to deal with a lot of video streams coming from different camera feeds. Such optimizations would be in terms of their performance with delays. Then comes data security issues as it will contain uploaded images and recorded videos which will require end encryption and access control.

1.6 Assumptions

- The system will rely on facial recognition based on facial features and expressions as a reliable identification method of missing persons in different video footage.
- It is expected that the automated detection and alert systems of the system would appreciably assist law enforcement in reducing their response time to increase the chances of recovering missing persons.
- The platform will operate in an environment with adequate CCTV camera coverage and high-resolution footage, ensuring reliable and uninterrupted real-time processing and monitoring.
- Authorized personnel, such as law enforcement officials, will receive appropriate training on how to interpret and utilize the system's alerts, match confidence levels, and filtering tools effectively to support investigative efforts.

1.7 Societal / Industrial Relevance

The project holds significant societal relevance as it aims to improve the efficiency and accuracy of missing person identification, addressing a critical need in public safety. By providing a system that automates the identification of missing individuals through advanced facial recognition, the project offers a powerful tool to aid law enforcement, enabling faster response times and increasing the likelihood of safe recoveries. The system can be applied in a wide range of public settings, including schools, airports, transit hubs, and large events, helping communities prevent prolonged disappearances and enhancing the overall security infrastructure.

In the security and surveillance industry, this project contributes to the advancement of AI-driven monitoring solutions, setting a new standard for the use of face recognition in high-density environments. The Missing Person Identification System's scalable and automated approach can influence the development of future technologies, supporting more proactive and efficient public safety measures in urban areas and beyond.

1.8 Organization of the Report

This report is organized as follows:

- **Chapter 1** provides an introduction to the project, including its aims, problem definition, scope, objectives, and relevance to society and industry.
- **Chapter 2** discusses the background and literature review, examining existing research and technologies related to automated facial recognition and missing person identification.
- **Chapter 3** presents the design and implementation process, describing the development of each system module, from video processing to face recognition, and the integration of these components into a cohesive platform.
- **Chapter 4** presents the design and implementation process, describing the development of each system module, from video processing to face recognition, and the integration of these components into a cohesive platform.
- **Chapter 5** covers the results and analysis, evaluating the system's performance across different conditions, such as lighting variations and video quality, and measuring its accuracy in identifying missing individuals.
- **Chapter 6** provides the conclusions and future scope of the project. It summarizes the key findings, reflects on the system's practical impact, and proposes enhancements such as real-time optimization, improved generalization, integration with law enforcement, mobile deployment, privacy safeguards, and ethical AI practices.

1.9 Conclusion

With increased urbanization, present-day cities are characterized by ever-growing complexity and density of human activities, rendering impossible all their counter public safety efforts put forth to track down missing persons. Hence, this project attempts to develop an effective Artificial Intelligence-based and facial recognition-based Missing Person Identification system that would help in identifying missing individuals at a faster pace and with greater accuracy. Much of the automation and processing of video-real-time face matching with alert generation-have minimized manual monitoring and have therefore enabled law enforcement to act swifter, more accurate, within high-traffic, congested areas.

This system also brings future replacement cost-saving methods and extends to scoping scalable tools for various settings for augmenting surveillance networks by way of an intervention into the existing public safety infrastructure. The relevance of this project in the future public safety initiative is an immediate solution to a very urgent need for an efficient, reliable identification system. With modern-day AI and improvements in video processing, the project sets the groundwork for future innovations that will further enhance the efficiency of the process of locating missing persons and makes communities more secure and responsive in emergencies.

Chapter 2

Literature Survey

2.1 A Real-Time Framework for Human Face Detection and Recognition in CCTV Images[1]

The paper proposes a real-time system that detects and recognizes human faces using CCTV footage. It covers issues like variations in lighting, scaling, rotation, and cluttered backgrounds. The system combines a machine learning and deep learning pipeline, such as feature extraction and classification algorithms, in order to achieve a high level of surveillance technology with the least human involvement.

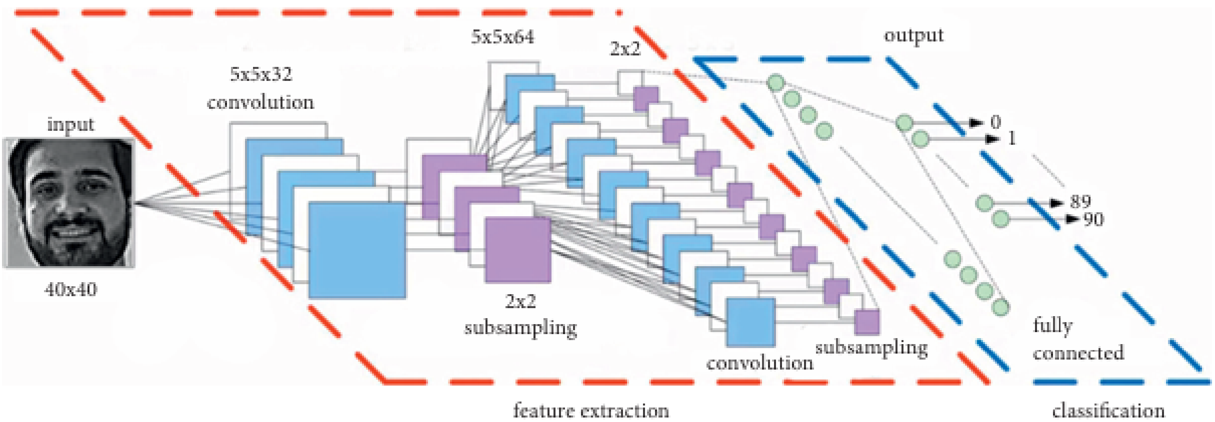


Figure 2.1: The architecture of CNN.

The proposed framework consists of four main phases: image acquisition, preprocessing, face detection, and face recognition.

- **Image Acquisition:** Images are captured continuously from CCTV cameras and labeled for classification. A dataset of over 41,320 images across 90 individuals was developed.

- **Preprocessing:** This involves grayscale conversion and edge detection using Gaussian and Sobel filters to enhance image quality for further processing.

- **Face Detection:** The Viola-Jones algorithm is utilized to detect and localize faces. The detected regions are extracted and resized to 40×40 pixels.
- **Face Recognition:** Various methods, including Principal Component Analysis (PCA), K-Nearest Neighbor (KNN), Random Forest, Decision Tree, and Convolutional Neural Networks (CNN), are implemented for feature extraction and recognition.

The feature extraction phase uses PCA to reduce image dimensions and identify key facial features. The recognition phase compares the performance of several classification techniques:

- **K-Nearest Neighbor (KNN):** Accuracy varies based on the number of eigenvectors and distance metrics. A maximum accuracy of 95% was achieved with Manhattan distance and 5 eigenvectors.
- **Random Forest:** Achieved up to 93.2% accuracy with five eigenvectors.
- **Decision Tree:** Exhibited lower accuracy compared to other methods.
- **Convolutional Neural Networks (CNN):** Outperformed other techniques, achieving up to 97.5% accuracy with an 80:20 training-testing split.

Results: The CNN approach yielded the highest accuracy, demonstrating its superior capability in face recognition. The performance of PCA with traditional machine learning classifiers was comparatively lower but still effective under certain conditions.

Contributions:

- Creation of a dataset with over 40,000 images captured under diverse conditions.
- A thorough comparison of classical machine learning and deep learning techniques for face recognition.
- A robust framework for automating face recognition in real-world CCTV applications.

Future Work: The authors plan to extend the system to recognize multiple faces in live-streaming videos, enabling a comprehensive real-time security solution.

2.2 Methodology for Facial Electromyographic (EMG) Signal Analysis in Emotion Recognition 2024[2]

This study investigates emotion recognition by analyzing facial electromyographic (EMG) signals, a precise technique for capturing muscle activity linked to specific emotional expressions. EMG captures nuanced, real-time data, especially when electrodes are positioned per the Facial Action Coding System (FACS). Here, EMG data was gathered for emotions including joy, sadness, surprise, and anger from multiple participants, with signal power features used to train classifiers and evaluate the accuracy of emotion recognition.

2.2.1 Experimental Design and Signal Collection

Experiment Setup

The study used the g.Tec g.USBAmp bioelectrical amplifier and eight active electrodes to capture EMG signals from 16 participants. Electrodes were positioned according to the FACS guidelines, targeting muscles involved in emotional expressions. Participants, aged 20–57, were instructed to express emotions naturally and subtly. The setup included a relaxed environment with brief instructions to ensure uniform signal collection across participants.

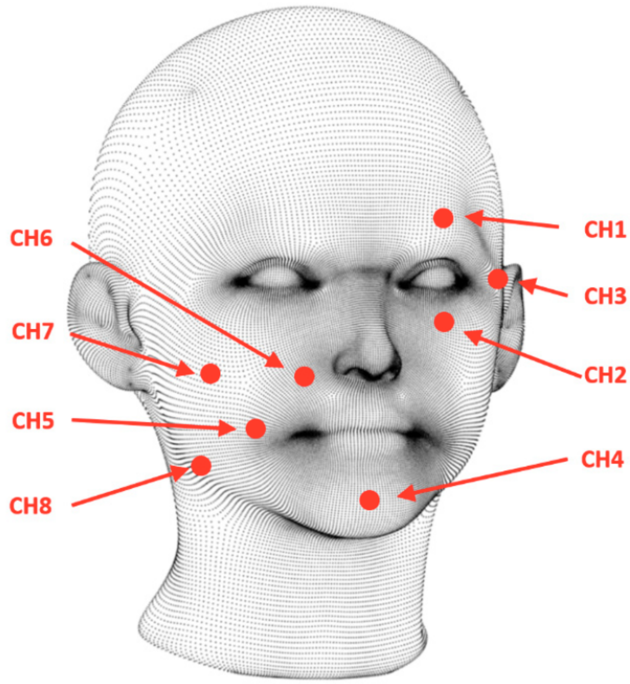


Figure 2.2: Placement of EMG electrodes on the face.

Signal Acquisition and Data Collection

Each participant acted out emotions for 30-second intervals. During this time, the eight electrodes recorded muscle activity across the face, targeting expressions associated with joy, sadness, anger, surprise, disgust, and fear, along with a neutral state. Data segments were saved as separate instances for each participant, creating a comprehensive database of EMG signals for analysis.

2.2.2 Signal Processing and Feature Extraction

Signal Preprocessing

The collected EMG signals underwent a preprocessing phase using OpenViBE software to improve signal quality. High-pass (0.1 Hz) and low-pass (1000 Hz) filters were applied, and a 50 Hz notch filter removed power grid noise. Additionally, Butterworth notch filters were applied at harmonic frequencies (100 Hz, 150 Hz, etc.) to eliminate further interference. No other preprocessing steps were applied, maintaining the natural variability of the muscle activity.

Feature Extraction

To simplify the classification process, mean signal power in the time domain was chosen as the primary feature for emotion recognition. This measure was calculated over each 1/4-second window for each channel. The mean power P_x is defined as:

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2 \tag{2.1}$$

where:

- N is the number of samples in the window,
- $x[n]$ is the amplitude of the EMG signal at sample n.

This measure provided a stable, interpretable feature indicating muscle activity levels linked to specific emotions. The extracted feature formed the basis for training the classification models.

2.2.3 Emotion Classification Models

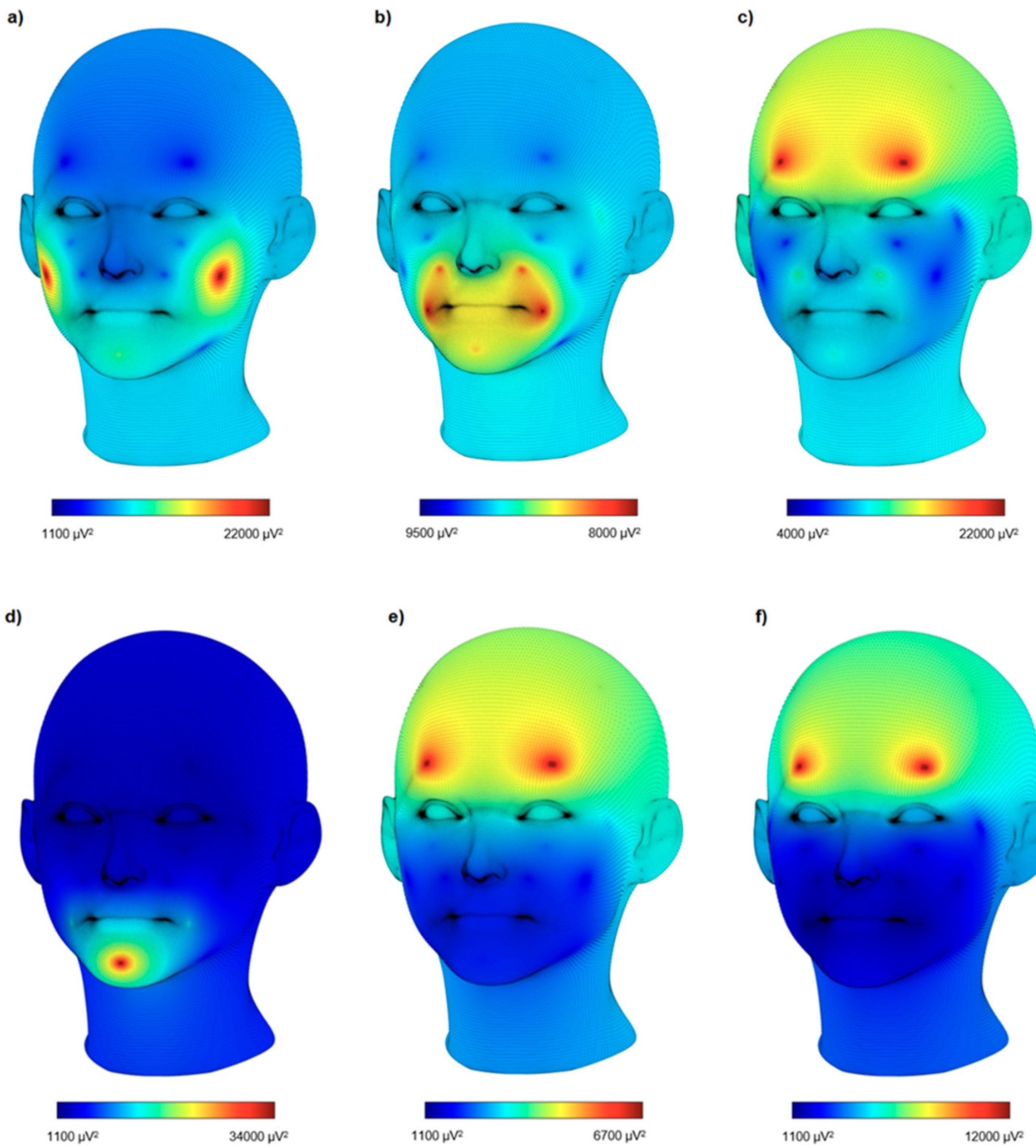


Figure 2.3: Distribution of signal power while expressing (a) joy, (b) disgust, (c) anger, (d) sadness, Figure 3. Distribution of signal power while expressing (a) joy, (b) disgust, (c) anger, (d) sadness, (e) fear, and (f) surprise..

2.2.4 Classification Algorithms

To classify emotions, four algorithms were used: k-nearest neighbors (KNN), support vector machines (SVM) with linear and cubic kernels, and linear discriminant analysis (LDA). These algorithms were evaluated using 10-fold cross-validation, a method that splits data into training and testing sets across ten iterations, reducing model overfitting. Each classifier's accuracy in recognizing the seven emotional states provided insights into the best algorithm for EMG-based emotion recognition.

2.2.5 Classification Approaches

Two main approaches were tested for classifying emotions:

1. **Subject-Dependent Approach**: Each model was trained and tested on data from individual participants to assess performance when tailored to specific users.
2. **Subject-Independent Approach**: Models trained on data from a group of participants were tested on new users, simulating real-world applications where emotion recognition would generalize to new individuals.

2.2.6 Results and Analysis

Subject-Dependent Classification

The proposed generalization to new users is called subject-independent approach classifiers. Of the classifiers applied in this research, KNN has achieved an average accuracy of 67.5 % on this setup, demonstrating that models have some agreement in EMG signals for different people but have less performance than in the subject-dependent approach. Classifiers suffer particularly in separating emotions similar to anger and disgust from each other and indicate further research to improve generalizability.

Subject-Independent Classification

The subject-independent approach aimed at generalizing classifiers to new users. KNN achieved the highest average accuracy in this setup, at 67.5%, indicating that while models captured some consistency in EMG signals across users, their performance was lower compared to the subject-dependent approach. Classifiers faced challenges, especially in distinguishing similar emotions like anger and disgust, underscoring the need for further research to improve generalizability.

2.2.7 Discussion

Comparison with Other Studies

Findings of this study outperform most existing researches regarding accuracy in emotion recognition, especially with the subject-dependent approach. For instance, KNN recorded above 96 % accuracy while other studies that used fewer electrodes or simpler classifiers reported lower performance. This reinforces that a combination using strategically placed electrodes and robust classifiers can enhance emotion recognition.

2.2.8 Practical Implications and Limitations

The study demonstrates that facial EMG provides a reliable means of emotion recognition, especially in subject-dependent settings. However, the variability in electrode positioning and individual skin properties introduces challenges, particularly in subject-independent models. Expanding the study to a larger and more diverse sample and exploring adaptive models could help mitigate these limitations.

2.2.9 Conclusion

In summary, this research shows that facial EMG signals can effectively differentiate emotions, with KNN and SVM models achieving high accuracy in subject-dependent setups. The mean power feature was simple and robust, contributing to efficient emotion recognition. Future studies should focus on refining subject-independent models and increasing sample diversity to enhance generalizability for broader applications. [?]

2.3 A Novel Technique for Automated Concealed Face Detection in Surveillance Videos[3]

The study proposes a technique to detect concealed faces in surveillance footage, addressing the need for reliable face detection in high-security areas. The method uses **color-space transformations**, **contour detection**, and **skin patch clustering** to accurately detect faces, even when partially covered. The approach is structured in three stages: **Human Detection**, **Head and Shoulder Contour Extraction**, and **Skin Patch Detection and Analysis**.

2.3.1 Methodology

The methodology consists of three primary stages: **Human Detection**, **Head and Shoulder Contour Extraction**, and **Skin Patch Detection and Analysis**.

Human Detection

Object Detection: The system continuously analyzes frames for new objects entering the scene. Height and width measurements are applied to identify potential human figures, isolating them based on defined thresholds.

Head and Shoulder Contour Extraction

When an object is classified as human, the system extracts the **head and shoulder** region to narrow down the search for skin patches.

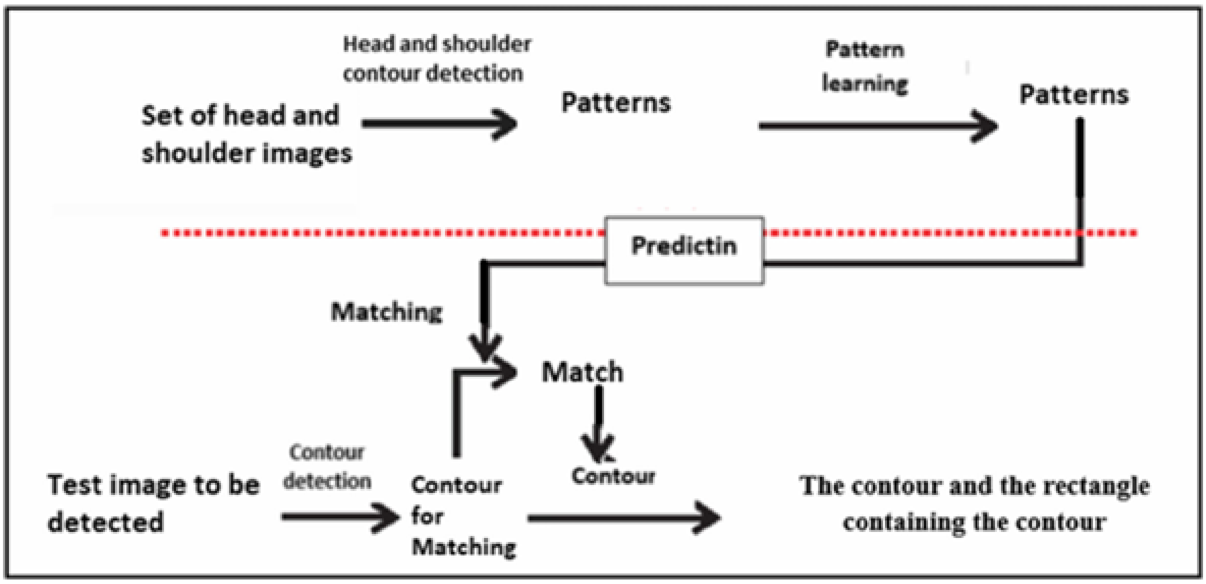


Figure 2.4: Head and shoulder training and detection

Contour Detection: Contour detection isolates the *head and shoulder* area using edge detection and shape analysis, identifying boundaries for the targeted region.

Feature Mapping: Key contour points on the head and shoulders are mapped to separate this region from the body, improving accuracy in locating skin patches.

Skin Patch Detection and Analysis

The detected head and shoulder region is analyzed for skin patches using a **hybrid skin detection model** that combines RGB and YCbCr color spaces.

Color-Space Transformation:

- **RGB Color Space**: The system uses normalized RGB color space to reduce dependence on lighting conditions, making detection adaptable to varied illumination.
- **YCbCr Color Space**: The YCbCr model separates chrominance (Cb, Cr) from luminance (Y), enhancing skin-tone distinction.

Skin Patch Clustering: The head and shoulder area is divided into small regions or patches, each analyzed to determine if it is a skin or non-skin region based on RGB and YCbCr thresholds.

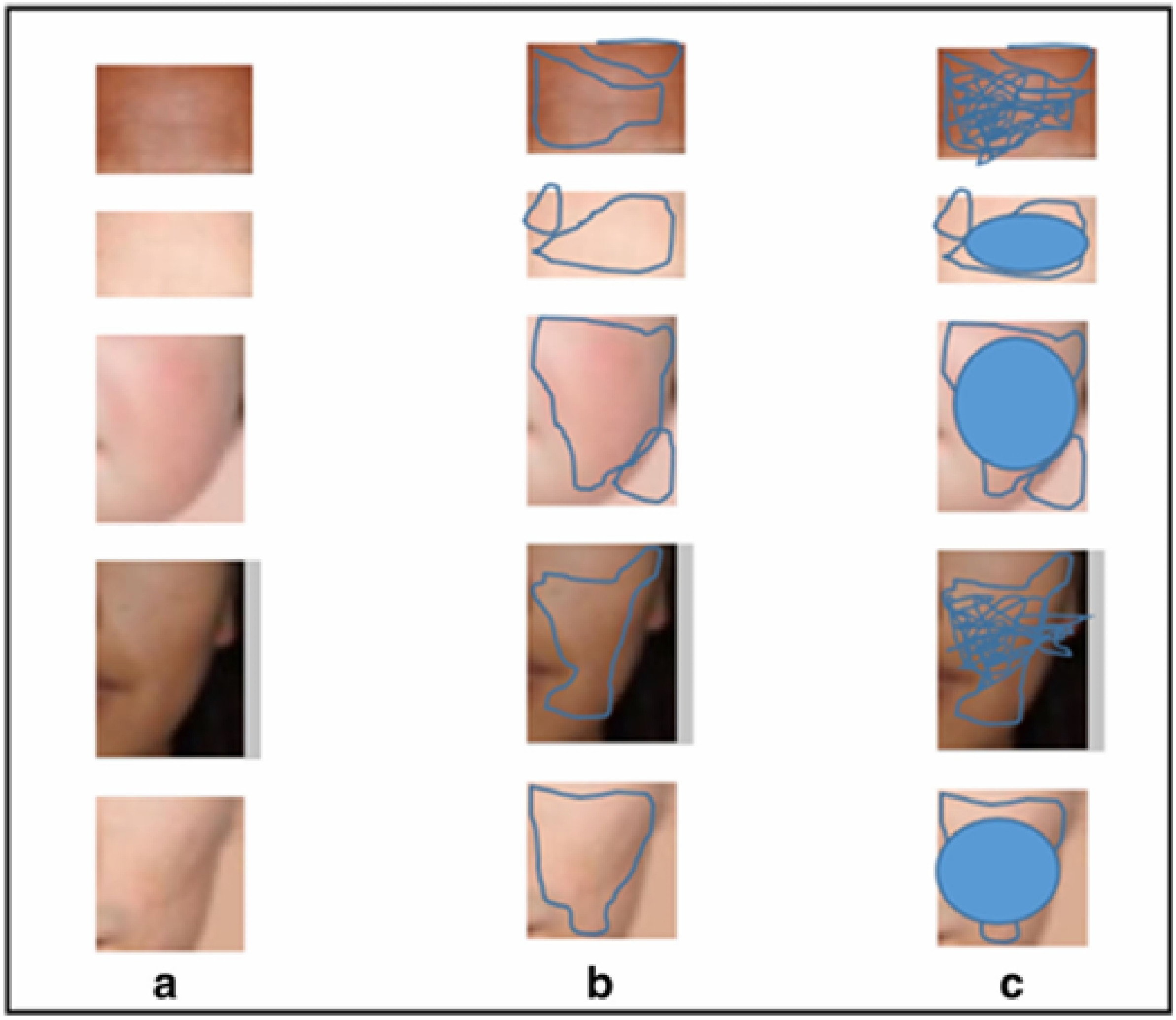


Figure 2.5: a. Patches of skin b. Clustering c. Concealing (dataset TAN)

Skin Detection Criteria: A patch is considered skin if it meets both RGB and YCbCr skin-tone criteria:

- **RGB Criterion**:

$$\frac{R}{R+G+B} > 0.36 \quad and \quad \frac{G}{R+G+B} > 0.28 \tag{2.2}$$

- **YCbCr Criterion**:

$$77 \leq Cb \leq 127 \quad and \quad 133 \leq Cr \leq 173 \tag{2.3}$$

If a patch meets these conditions, it is marked as skin; otherwise, it is classified as non-skin.

Concealed Patch Counting: For each head and shoulder region, the algorithm counts the number of non-skin patches. If more than half of the patches (i.e., $n/2+1$) are non-skin, the face is identified as concealed.

Binary Decision Rule: The system classifies the face as concealed or not using the following rule:

$$Concealed - Face = \{\ TrueifNon - SkinPatches > \frac{n}{2} + 1Falseotherwise \tag{2.4}$$

2.3.2 Summary of the Method

The technique uses the combination of color-space transformations and contour analysis to concealing face detection in surveillance video. It uses Hybrid RGB and YCbCr model for robust detection under changing light condition, therefore having a high security application..[3]

2.4 Improved face detection method via learning small faces on hard images based on a deep learning approach[4]

Face detection in complex environments, such as those with cluttered backgrounds, low resolution, and small faces, remains a challenging task in computer vision. The study *"Improved Face Detection Method via Learning Small Faces on Hard Images Based on a Deep Learning Approach"* addresses these challenges by proposing a deep learning-based solution that enhances face detection accuracy for small faces.

2.4.1 Challenges in Small Face Detection

Detecting small faces becomes very hard because their pixel representations fail to capture important features like eyes, nose, and mouth. The standard models either classify small faces incorrectly or entirely ignore them in environments with a lot of noise. The method proposed overcomes these limitations by integrating multi-scale feature extraction with training through robust techniques.

2.4.2 Multi-Scale Feature Pyramids

Multi-scale feature pyramids are included into the Model to increase the detection of faces with very different sizes. Each image feature is present in a different resolution so that small faces low on detail are able to be represented properly in the feature space.

2.4.3 Hard Negative Mining

The robustness of a model significantly improved with the inclusion of hard negative mining methods in the training process. Thus, the method helps mitigate false positive occurrences in small face detection tasks by concentrating on hard nonface areas.

2.4.4 RetinaNet-Based Architecture

Using the RetinaNet framework as a basis, this research paper utilizes a ResNet backbone to take advantage of the hierarchical feature extraction capabilities of ResNet. The combination of high- and low-resolution feature maps enables the model to achieve the efficiency-detector accuracy balance.

2.4.5 Performance on Standard Datasets

The method is tested using WIDER FACE and FDDB datasets, and it has a clear advantage in terms of precision, recall and F-measure compared to the traditional methods. The results show that it generalizes well across different ways and difficult real-world situations.

2.4.6 Summary

The work offers a very powerful and efficient technique for face detection. The study emphasizes the small faces in difficult situations. Using multi-scale feature extraction, hard negative mining, and the RetinaNet architecture, this approach deals with the major shortcomings of conventional face-detection systems and contributes to the advancement of disciplines such as surveillance, biometric recognition, and video analytics.

2.5 Upsampling Real-Time, Low-Resolution CCTV Videos Using Generative Adversarial Networks[5]

2.5.1 Methodology

The methodology in this paper focuses on a dual-branch GAN (Generative Adversarial Network) framework to upscale real-time, low-resolution CCTV videos. The approach integrates spatial and temporal aspects for improved high-resolution video generation.

Dual Generators: Spatial and Temporal Generators

Spatial Generator (GS)

The spatial generator processes low-resolution frames and learns detailed spatial features to produce initial high-resolution feature maps. It uses **Reversible Residual Blocks (RRBs)**, which efficiently learn spatial differences between low and high-resolution frames without significant memory overhead.

Reversible Residual Blocks (RRBs): Each RRB has a forward and reverse computation, allowing reconstruction of each layer's activation from the next layer's activation, reducing memory requirements. The RRB uses an additive coupling mechanism:

$$b_1 = a_1 + R(a_2) \quad and \quad b_2 = a_2 + F(a_1) \tag{2.5}$$

where:

- a_1 and a_2 are input features,
- b_1 and b_2 are output features,
- R and F are forward and reverse residual functions.

The reverse calculation for activations is:

$$a_1 = b_1 - R(b_2) \quad and \quad a_2 = b_2 - F(b_1) \tag{2.6}$$

This architecture enables G_S to preserve fine-grained spatial details while limiting memory usage, which is crucial for handling large datasets in video processing.

Temporal Generator (GT)

The temporal generator maintains continuity across video frames by using **Convolutional Gated Recurrent Units (ConvGRU)** to handle the temporal dynamics of video frames. The ConvGRU operations for frame-to-frame consistency are:

$$u_t = \sigma(W_u * x_t + M_u * h_{t-1} + b_u) \tag{2.7}$$

$$r_t = \sigma(W_r * [u_t; x_t] + M_r * h_{t-1} + b_r) \tag{2.8}$$

$$h_t = u_t \circ h_{t-1} + (1 - u_t) \circ \sigma(W_h * [x_t; r_t \circ u_t] + M_h(r_t \circ h_{t-1}) + b_h) \tag{2.9}$$

where:

- u_t and r_t are update and reset gates,
- h_t is the updated hidden state at time t,
- W and M represent weight matrices for the operations.

Sparse Matrix Fusion: The outputs from both G_S and G_T are fused using a sparse matrix to prevent overfitting while ensuring effective feature selection. The fused feature map F_{map} is calculated as:

$$F_{map} = F_{concat} \times M_S \tag{2.10}$$

where F_{concat} are the combined spatial and temporal features, and M_S is the sparse matrix used for fusion.

2.5.2 Dual Discriminators: Spatial and Temporal Discriminators

Spatial Discriminator (DS)

The spatial discriminator evaluates individual frames to verify if they are real or generated. It uses a ResNet-based architecture and a binary classification layer for this purpose.

Temporal Discriminator (DT)

To ensure temporal consistency across frames, D_T leverages **R(2+1)D Convolutions**. This approach splits the 3D convolutional layer into a 2D spatial and 1D temporal convolution, reducing computational complexity and improving the training process.

For a frame sequence $\{x_t\}$, the temporal feature extraction is represented as:

$$D_{temporal} = \sum_{i=1}^{n} R(2+1)D(x_i) \tag{2.11}$$

where n represents the number of frames considered.

2.5.3 Loss Functions

Three key loss functions are used to train the model, combining pixel-level accuracy, feature-level similarity, and adversarial training.

Content Loss (Mean Squared Error)

The content loss is calculated as the Mean Squared Error (MSE) between generated high-resolution frames and the ground truth high-resolution frames. This ensures structural similarity at a pixel level.

$$L_{con} = MSE(\theta) = \frac{1}{T_s} \sum_{i=1}^{T_s} \|HR_i - M(LR_i, \theta)\|^2 \tag{2.12}$$

where:

- T_s is the number of samples,
- HR_i and LR_i are the high and low-resolution frames,
- M is the mapping function between low and high-resolution frames.

Adversarial Loss

The adversarial loss encourages the generator to create frames that can deceive the discriminators, enhancing realism. It is defined as:

$$L_{adv} = -\log(D_\psi(S|LR)) \tag{2.13}$$

where S is the generated synthetic frame, and LR are the low-resolution input frames.

Perceptual Loss

To focus on feature-level similarities, perceptual loss is calculated using a pre-trained VGG-19 network. This loss measures differences in feature representations, improving texture and fine details.

$$L_{perc} = \frac{1}{W_n H_n} \sum_{i=1}^{W_n} \sum_{j=1}^{H_n} (\phi_n(HR)_{i,j} - \phi_n(G(LR))_{i,j})^2 \tag{2.14}$$

where:

- W_n and H_n are the width and height of the feature map,
- ϕ represents activation of a specific layer.

Total Loss Function

The total loss L used for optimizing the generator is a weighted sum of the content, adversarial, and perceptual losses:

$$L = L_{con} + \lambda_1 L_{adv} + \lambda_2 L_{perc} \tag{2.15}$$

where λ_1 and λ_2 are balancing coefficients.

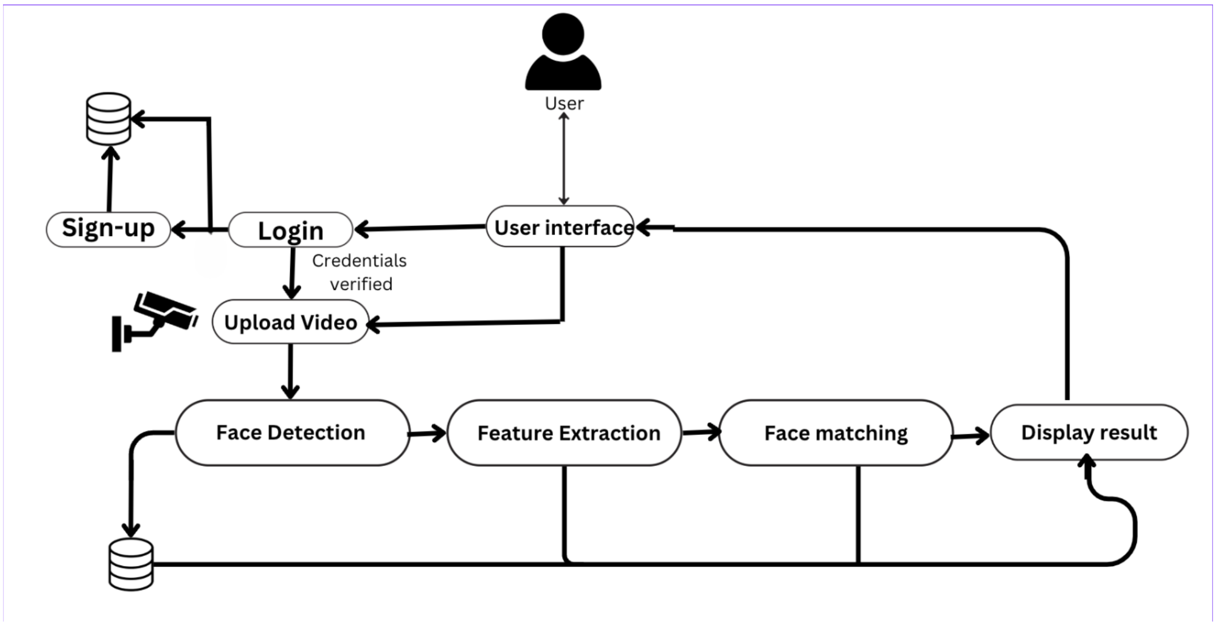
2.5.4 Summary

In short, introduces a dual-branch GAN framework to upscale low-resolution CCTV videos, integrating both spatial and temporal enhancements. The spatial generator captures fine details using Reversible Residual Blocks (RRBs), while the temporal generator maintains frame continuity with ConvGRU, combining outputs through sparse matrix fusion. Dual discriminators—spatial and temporal—ensure frame quality and realistic temporal coherence. The model is trained with content, adversarial, and perceptual losses, achieving precise detail and consistency, making it effective for improving CCTV footage resolution.

Chapter 3

System Design

3.1 System Architecture



(a) A high-level system architecture diagram that highlights the key components.

3.2 Component Design

Processing Input Video

Video Ingestion

The system is designed to support video input formats such as MP4, AVI, and MKV. Using libraries like OpenCV and FFmpeg, it can effectively read and process video frames.

Video upload handling

The user uploads a video along with an image through a web form. The backend handles the uploaded files using **request.FILES**, and saves them to designated folder using Django's **default_storage**. Once the video is saved, it's accessed using OpenCV (cv2.VideoCapture) and processed frame-by-frame.

Marking on Timeline

- **Detection Logging:** Timestamps are recorded for frames where individuals are detected and identified.
- **Time Interval Representation:** These timestamps are grouped into readable time intervals, such as `'00:02:15 - 00:02:30'`, to simplify report creation and interpretation.
- **Results Exportation:** The system can save the processed data onto a data path allowing for easy integration into analysis workflows or other systems.

Conclusion

These processes enable accurate video analysis and provide well-organized outputs suitable for downstream applications.

3.2.1 Image Pre-processing

This section describes the role of image pre-processing module.

- **Reference image handling:**
  - Input image handling: Store the input images in various formats such as JPEG, PNG, BMP, TIFF.
- **Image enhancements:**
  - Detected face images must be resized to match the input dimensions required by DeepFace (typically 224×224 pixels for most models).
  - Resizing ensures that the face recognition model receives images of a consistent size, improving accuracy and computational efficiency.
  - This step helps avoid distortions that may arise due to varying image resolutions in CCTV footage.
  - Common libraries for resizing include OpenCV (cv2.resize) and PIL (Python Imaging Library).
- **Image Alignment:**
  - Landmark based image alignment is utilized for identifying similar images in the system.
  - Used to improve recognition capabilities of the system.
  - Dlib's facial landmark detection can detect 68 key points on the face.
  - Affine transformations (using OpenCV) can adjust the orientation to standardize the alignment.
- **Image Validation:**
  - Validates the quality of image.
  - Verifies the reference images are distinct and usable for recognition tasks.
  - Brightness and Contrast Normalization: Adjust illumination to maintain uniformity.

3.2.2 YOLO-Based Face Detection

Components

- **YOLOv8 (face.pt):**
  - A highly optimized deep learning-based object detection model used for face detection.
  - Processes images at multiple scales, ensuring accurate detection of faces in varying sizes and lighting conditions.
- **DeepFace for Recognition:**
  - A robust face recognition framework leveraging deep learning techniques for extracting facial embeddings.
  - Supports multiple backbone models like FaceNet, ArcFace, and VGG-Face to enhance feature extraction accuracy.
- **Model Files:**
  - `YOLOv8 face.pt`: Pre-trained model for face detection.
  - `DeepFace pre-trained models`: Used for face embedding extraction and matching.

Key Features

- **Input Preprocessing:** Extracted frames are converted to grayscale, resized, and normalized to match DeepFace model input dimensions.
- **Face Detection:** Identifies multiple faces in real time using YOLOv8 and extracts bounding boxes.

3.2.3 Real-Time Face Detection Flow

1. **Frame Capture:** Captures frames from CCTV footage at predefined intervals (e.g., every 10 frames).
2. **Preprocessing:**

- Frames are converted to grayscale for consistency.
- Noise reduction and contrast normalization techniques are applied.
- Faces are aligned based on key landmarks to improve feature extraction.

3. **Inference:** The processed frames are passed through YOLOv8 for face detection.
4. **Face Embedding Extraction:** DeepFace extracts 128-dimensional embeddings from detected faces.
5. **Face Matching**
   Identification: Extracted features are compared against uploaded images for recognition.
6. **Post-processing:** Bounding boxes are drawn around detected faces with confidence scores.

Optimizations for Real-Time Performance

Objective Improve processing efficiency for real-time face detection and recognition.

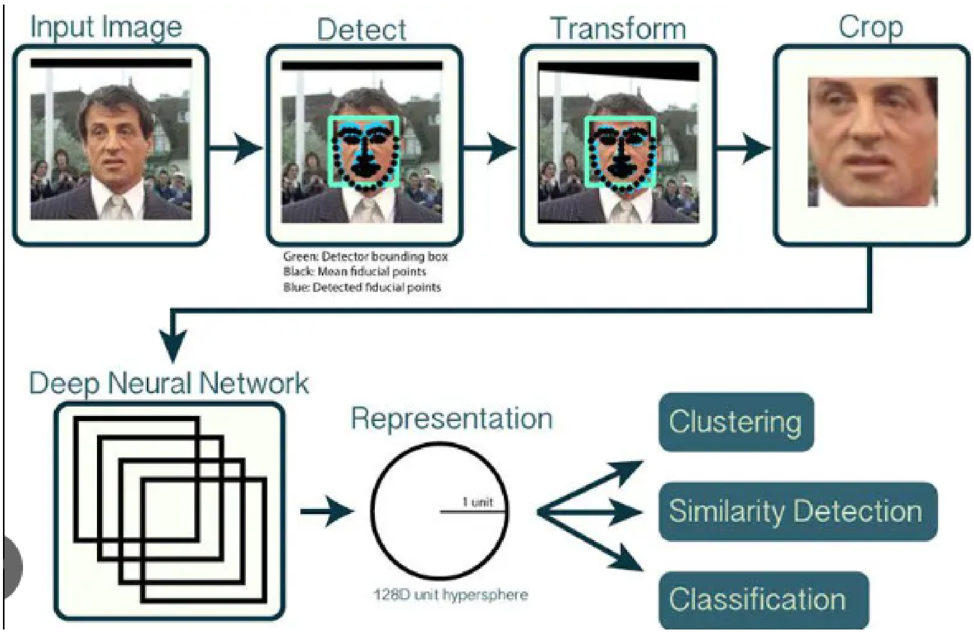
Techniques

- **Frame Skipping:** Processes every 10th frame from CCTV footage to reduce computational overhead while maintaining accuracy.
- **Input Resizing:** Detected face images are resized to match DeepFace input dimensions, ensuring faster feature extraction and comparison.
- **Face Alignment:** Uses facial landmark detection to standardize face orientation before feature extraction.

Outcome

- Enhanced real-time accuracy compared to traditional face detection techniques.
- Capable of detecting, identifying, and logging multiple faces per frame efficiently.

[HT]



(b) Real-time face detection

3.2.4 Face Matching Module

This section outlines the components of the Face Matching Module, which matches detected faces with a database of missing person.

Feature Extraction

Objective: Extract unique facial features as numerical vectors.

Process:

- Uses facenet to convert facial images into 128-dimensional feature vectors.
- Normalize inputs for consistent extraction.

Outcome: Generates standardized feature vectors for further matching.

Similarity Matching

Objective: Compare feature vectors to identify potential matches.

Process:

- Use Euclidean distance and cosine similarity to calculate match scores.

- Rank matches based on similarity.

Outcome: Produces a ranked list of potential matches.

Classification

Objective: Refine matches using machine learning.

Process:

- Use SVM to classify matches and KNN to identify nearest neighbors.
- Combine results for better accuracy.

Outcome: Outputs a high-confidence match or no-match result.

Threshold Evaluation

Objective: Validate matches using similarity thresholds.

Process:

- Accept matches exceeding the threshold.
- Flag ambiguous matches for manual review.

Outcome: Ensures precise and reliable match results.

Output Generation

Objective: Provide actionable results.

Process:

- Display match details and trigger alerts for matches.
- Log matched faces for future reference.

Outcome: Enables quick response and data logging.

Error Handling

Objective: Ensure robustness.

Process:

- Retry database connections and flag invalid inputs.
- Log errors for debugging.

Outcome: Maintains system reliability and operational stability.

3.3 Input and Output Specifications

3.3.1 Input

- High-resolution images of missing persons in formats such as JPEG, PNG, BMP, or TIFF.
- CCTV video footage in formats such as MP4, AVI, or MKV.

3.3.2 Output

- Alerts containing person details for identified matches.
- Timestamps and video segments where matches are detected.

3.4 Algorithm Design

3.4.1 Video Processing Module

Objective

To process input video streams, extract relevant frames, and prepare them for subsequent face detection and recognition.

Algorithm

1. **Video Ingestion:** Utilize OpenCV or FFmpeg to load and process video streams efficiently.
2. **Frame Extraction:** Extract frames at a fixed interval (e.g., 2 FPS) to balance computational efficiency and detection accuracy.
3. **Video Upload processing:** Using django framework designate path to store the uploaded video file.
4. **Timestamp Logging:** Store timestamps of detected moving objects to facilitate quick reference during recognition.
5. **Data Export:** Save extracted frames and metadata (timestamps, detected objects).

Output

- Processed video frames with corresponding timestamps.
- Structured datasets containing extracted information for further face detection and recognition.

3.4.2 Image Preprocessing Module

Objective

To enhance input images and prepare them for accurate face detection and recognition.

Algorithm

1. **Input image handling:** Using Django framework handle storage of input image for referencing.
2. **Image Enhancement:** Apply noise reduction techniques such as Non-Local Means (NLM) and adjust brightness/contrast for poorly lit images. Apply unsharp masking or high-pass filtering for edge enhancement.
3. **Image Alignment:** Use Dlib to align images based on facial landmarks (e.g., eyes, nose, mouth).
4. **Validation:** Verify the quality and distinctiveness of input images to ensure accuracy in detection and matching.

Output

- Enhanced and aligned images ready for face detection.

3.4.3 Real-Time Face Detection Module

Objective

To detect faces in video frames or images in real time.

Algorithm

1. **Input Preprocessing:** Resize frames to 300×300 pixels and normalize pixel values using OpenCV's `cv2.dnn.blobFromImage()` function.

2. **Model Inference:** Use SSD with a ResNet-10 backbone for face detection. Load pre-trained weights (`res10 300x300 ssd iter 140000 fp16.caffemodel`).

3. **Bounding Box Generation:** Draw bounding boxes around detected faces with confidence scores above a predefined threshold.

4. **Post-Processing:** Extract detected face regions and store them for further analysis.

Output

- Detected face regions with bounding boxes.
- Confidence scores for each detection.

3.4.4 Face Matching Module

Objective

To match detected faces with a database of missing persons to aid in identification and recovery.

Algorithm

1. **Feature Extraction:** Utilize CNN models to convert face images into 128-dimensional feature embeddings. Normalize these embeddings to ensure consistency across different lighting and pose variations.

2. **Similarity Matching:** Compare extracted feature embeddings against stored feature vectors using distance metrics such as Euclidean distance and cosine similarity. Rank potential matches based on similarity scores.

3. **Classification and Decision Making:**

    - Apply an SVM classifier for high-confidence classification.

- Use the k-NN algorithm to refine borderline cases by analyzing neighboring feature vectors.

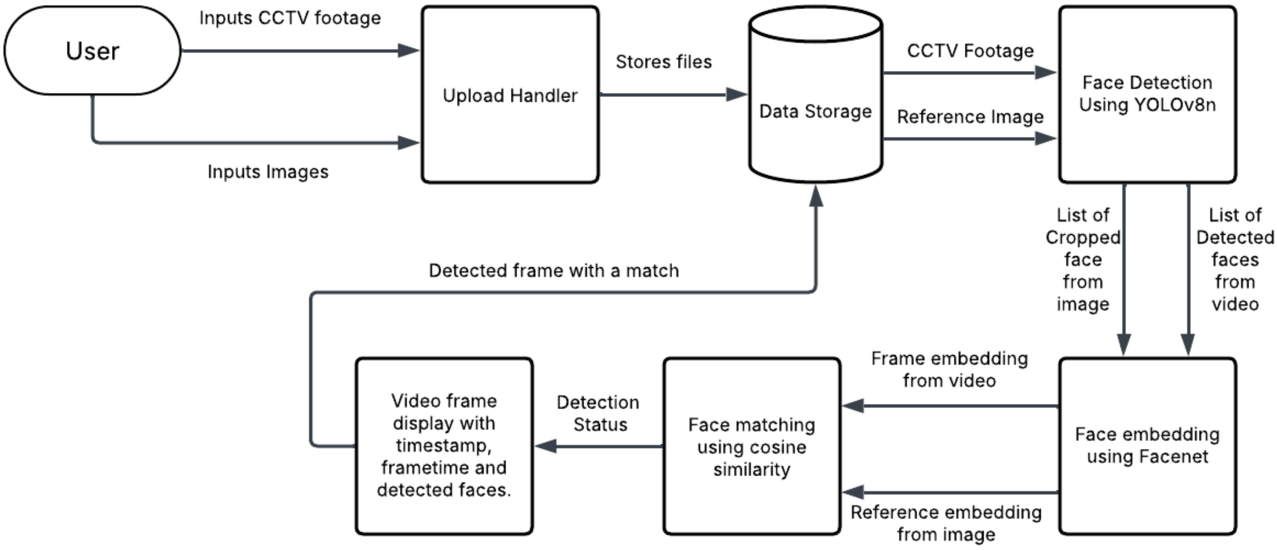
4. **Threshold Evaluation:** Define similarity thresholds to validate matches and flag low-confidence cases for manual verification.

5. **Alert Generation:** Notify relevant authorities upon identifying a high-confidence match and log unmatched faces for further analysis.

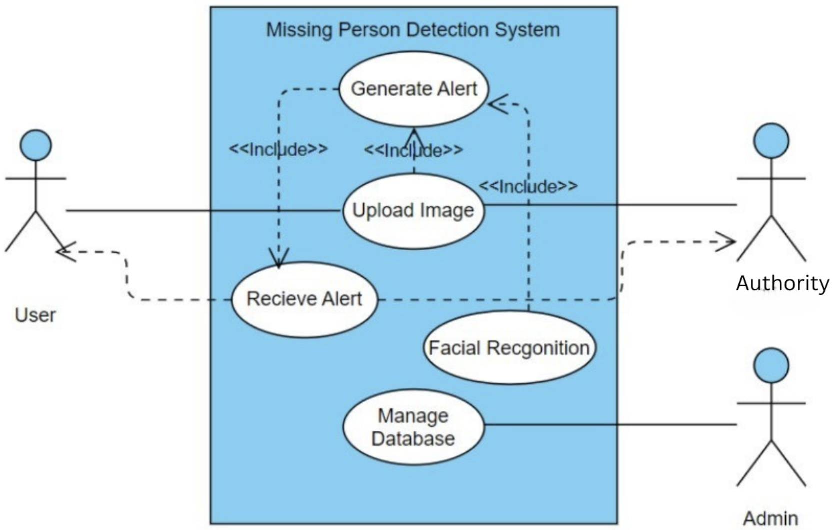
Output

- Automated alerts for identified matches, including associated details.
- Logs of unmatched detections for later review and potential inclusion in training updates.

3.5 Data Flow Diagrams (DFD)/ USE CASE diagram



(c) Data Flow Diagrams (DFD)



(d) USE CASE diagram

3.6 Tools and Technologies: Software/Hardware Requirements

This section outlines the essential software and hardware components required for the effective development and deployment of the project.

Software Requirements:

- **Python**: For backend development and algorithm implementation, enabling core system logic and computations.
- **HTML CSS**: For front-end development, ensuring an interactive and visually appealing user interface.
- **OpenCV**: For computer vision tasks like face and eye tracking, supporting real-time analysis.
- **TensorFlow/PyTorch**: For AI and deep learning models, enabling training and inference of neural networks.
- **YOLOv8n** : Utilized for advanced face detection, improving accuracy and robustness in real-time applications.
- **DeepFace**: A traditional machine learning-based approach for initial face and eye detection tasks.
- **Django**: For creating a web-based interface, enabling interaction with the system via a web browser.
- **Git**: Version control system for tracking changes to the codebase, facilitating collaboration and learning.
- **Visual Studio Code/PyCharm**: Integrated Development Environment (IDE) for coding and debugging.
- **pip**: Package management tool for handling Python dependencies.

Hardware Requirements:

- **Processor**: Intel Core i5 or equivalent, providing sufficient computational power for data processing.

- **RAM**: 16GB or more for smooth multitasking and handling large datasets.
- **Storage**: 512GB SSD for faster data access and storage of video files.
- **Camera**: An HD webcam for face and eye tracking which is very necessary in data acquisition in real time.
- **GPU**: 4GB Accelerated inference and processing for deep learning.
- **Operating System**: Windows 10/11 which incorporates the development tools and libraries for usage.
- **Networking**: Reliable and high-speed internet connections without interruption possible image upload and retrieval of data.

3.7 Module Divisions and work break down

This chapter gives an overview of the different modules incorporated in the system, specifying all the functionalities they perform and the technical implementations they adopt.

3.7.1 Video Processing Module

Objective

The objective of the video editing module is to extract frames from CCTV footage and process them for subsequent tasks.

Key Steps

- **Frame Extraction:** Capturing frames at regular intervals, ensuring sufficient data is available for analysis while minimizing redundancy.
- **Handle video input:** Using Django framework handle uploading of video footage.
- **Metadata Extraction:** Extracting timestamps and relevant metadata to facilitate the search and analysis of detected faces with associated timestamps.
- **Display video Output:** Display video output to the user.

Image Preprocessing Module

Objective

The objective of the image preprocessing module is to enhance the quality of images for more accurate detection.

Key Steps

- **Removal of Noise:** Eliminating unwanted noise from the images to ensure it does not interfere with face detection.
- **De-blurring:** Applying sharpening techniques to improve the clarity of faces in images, enhancing detection accuracy.

- **Normalization (Alignment):** Aligning faces in images to standardize their orientation and position for more consistent detection results.

- **Normalization (Lighting):** Standardizing variations in lighting, contrast, and brightness to ensure uniformity across images.

- **Input and output image handling:** Handle the storing and processing of input output images.

3.7.2 Face Detection Module

Objective: Mutual Detection of Frames and Faces

Key Steps

- **HOG (Histogram of Oriented Gradients):** Utilizes HOG features to identify faces by analyzing gradient changes and edge detection within images.

- **Dlib's Landmark Detection:** Detects key facial points, such as the eyes, nose, and mouth, which aid in precise alignment and subsequent processing.

3.7.3 Face Matching module

Objective: To match detected faces with stored images of missing person

Key Steps

- **Feature Extraction:** Utilize Convolutional Neural Networks (CNNs) to generate a 128-dimensional feature vector for each individual face.

- **Face Matching:** Compare the learned feature vectors against all stored feature vectors of missing person using algorithms like Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) to find the best match among all detected faces.

3.8 Key Deliverables

3.8.1 Input Processing

- **Multiple-Format Support:** The system supports the ingestion and processing of CCTV footage in various video formats and resolutions.
- **Preprocessing Enhancement:**
  - **Noise Reduction:** Reduce noise in video frames for improved clarity and processing.
  - **Image Enhancement:** Enhance the quality of images for better analysis.
  - **Upsampling:** Use techniques like Generative Adversarial Networks (GANs) or similar approaches to upscale low-resolution footage.

3.8.2 Person Identification

- **Face Recognition** Accurately detect faces using advanced algorithms and extract facial features with models such as ResNet-10 or YOLOv8.
- **Database Matching** Match facial embeddings against a database of missing persons. Provide confidence scores and ranked matches for identified individuals.

3.8.3 Timeline Generation

- **Timestamped Highlights** Extract video segments where the identified person appears, with precise timestamps (e.g., `"00:02:15 - 00:02:45"`).
- **Bounding Box Annotation** Annotate bounding boxes around individuals in relevant frames using tracking algorithms like DeepSORT for continuity.

3.8.4 Report Information

- **Visualization** Display video clips with highlighted sections and an interactive timeline for navigation.
- **Text Summary** Provide detection logs with timestamps and confidence scores for each identified individual.

- **Export Options** Export annotated video clips and reports in standard formats such as PDF and JSON for further use.

3.8.5 System Features

- **Scalability:** Capability to handle multiple video streams or bulk footage simultaneously.
- **Accuracy and Robustness:** Reliable identification even under challenging conditions such as occlusions or low lighting.
- **Real-Time Processing (Optional):** Alerts for live feeds when the person is detected in real time.

3.8.6 Security and Compliance

- **Data Security:** Ensured encryption and access control for footage and personal data.
- **Regulatory Compliance:** Adherence to local privacy and data usage laws.

3.8.7 Performance Metrics

- **Accuracy:** High true-positive rate with minimized false positives and negatives.
- **Processing Speed:** Efficient analysis, enabling near-real-time alerts or processing of bulk footage within hours.

3.9 Project Schedule

(e) Gantt Chart for the Project Phases

3.10 Conclusion

The modules which have been mentioned above embody a very advanced type of technology. They include, but are not limited to, video processing, image enhancement, and, most importantly, machine learning techniques. This altogether forms a fix designed to introduce an efficient identification for missing persons. Based on the facial recognition algorithms and certain other preprocessing methods, the system gives real-time face detection and matching for timely identification of missing children. With improvements, this technology could well mark a milestone in locating missing persons, thereby improving the safety and security of public space.

Chapter 4

System Implementation

This chapter focuses on the implementation phase of the project, detailing the proposed methodology, user interface design, implementation strategies, and concluding remarks.

4.1 Proposed Methodology/Algorithm

The implementation of our system follows a structured methodology to ensure efficient performance and high accuracy. The methodology consists of the following key steps:

1. **Data Collection:** The system processes reference images uploaded by school authorities or parents and extracts relevant frames from CCTV surveillance footage.

2. **Preprocessing:** Extracted frames are converted to grayscale, noise reduction techniques are applied, and brightness/contrast adjustments are made to enhance image quality.

3. **Face Detection:** YOLOv8 (face.pt) is used to detect and crop faces in CCTV frames for further analysis.

4. **Feature Extraction:** DeepFace extracts 128-dimensional face embeddings using models such as FaceNet, ArcFace, or VGG-Face for robust feature representation.

5. **Face Matching:** Extracted facial embeddings are compared with uploaded student images using Cosine Similarity or Euclidean Distance.

6. **Alert System:** If a student is identified, detection timestamps are logged, and notifications are sent to parents and school authorities.

4.2 User Interface Design

The system provides an intuitive and interactive user interface for seamless user interaction, designed based on the following principles:

- **Minimalistic Design:** Simplified UI to enhance usability and reduce cognitive load.
- **Interactive Elements:** Forms, buttons, and dropdowns ensure smooth user navigation.
- **Real-time Feedback:** The system validates user inputs and provides real-time suggestions.
- **Error Handling:** Meaningful error messages guide users in case of incorrect inputs or system failures.

The frontend is implemented using HTML, CSS, and JavaScript, ensuring accessibility across various devices. Figures 4.1 to 4.4 showcase different user interface screens of the application.

4.3 Implementation Strategies

A structured approach is adopted to ensure robustness and efficiency during system implementation. The major strategies employed are described below:

4.3.1 Technology Stack

The system leverages the following technologies:

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Django with Python
- **Machine Learning:** OpenCV, YOLOv8, DeepFace, TensorFlow/PyTorch for face detection and recognition

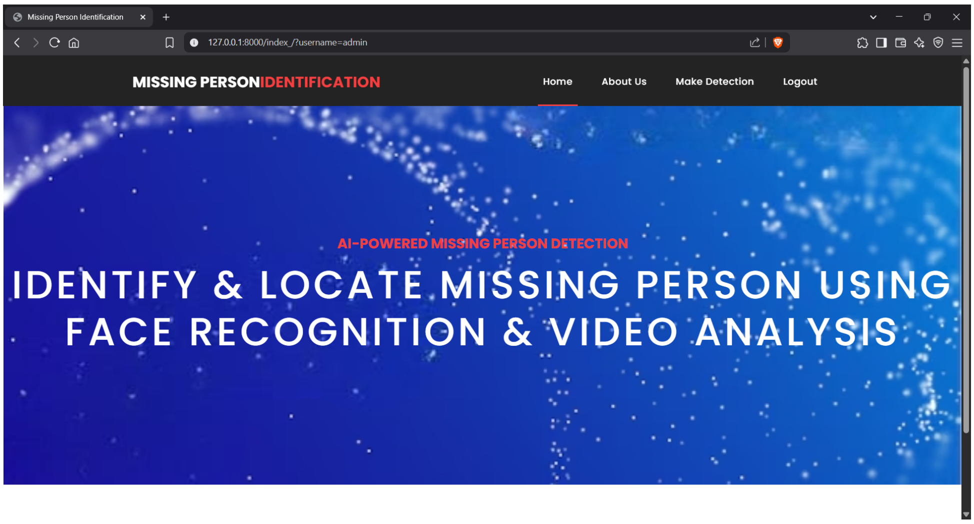


Figure 4.1: Home Screen

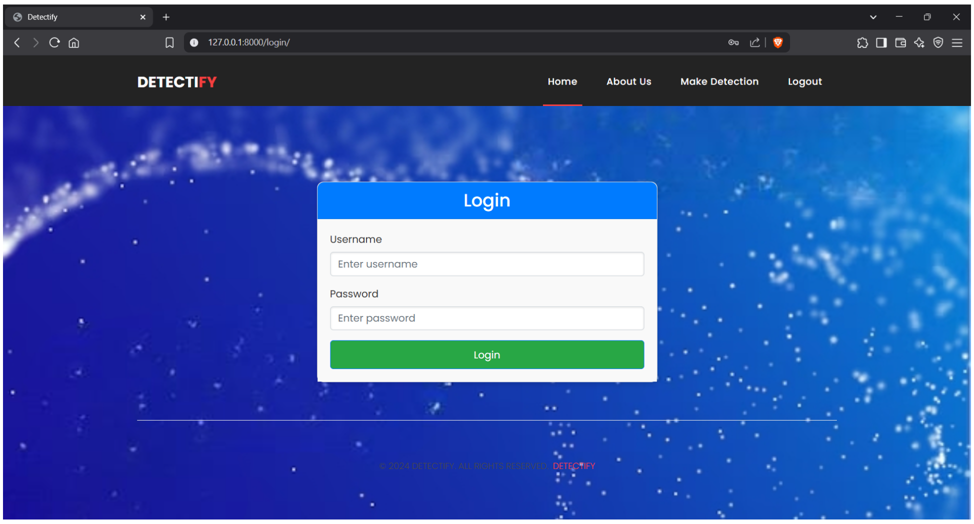


Figure 4.2: Login Page

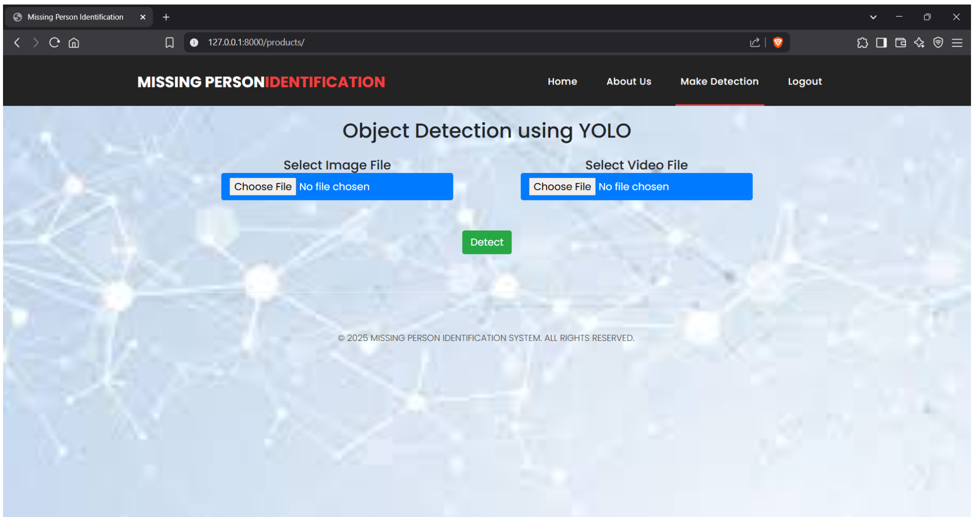


Figure 4.3: CCTV Upload Screen

Figure 4.4: Face Recognition Results Screen

Figure 4.5: Face Recognition Saved Results

4.3.2 Modular Development

The system is designed with a modular architecture to ensure scalability and maintainability:

- **Face Detection Module:** YOLOv8 detects faces in video frames.
- **Face Recognition Module:** DeepFace extracts features and performs face matching.

4.3.3 Security Measures

To ensure data security and prevent misuse, the system implements the following security measures:

- **Authentication:** Firebase Authentication secures user access.
- **Data Encryption:** Sensitive data is encrypted before storage.
- **SQL Injection Prevention:** Parameterized queries prevent malicious database attacks.

4.3.4 Performance Optimization

The system incorporates the following optimizations to enhance efficiency:

- **Frame Sampling:** Extracting frames at optimal intervals reduces processing load.
- **GPU Acceleration:** Face detection and recognition tasks leverage GPU-based computations.
- **Database Indexing:** Optimized indexing speeds up query execution.

4.4 Chapter Conclusion

This chapter provided an in-depth discussion on the system implementation, including the proposed methodology, user interface design, and various implementation strategies. The system is designed to be efficient, scalable, and secure while providing an intuitive user experience. The next chapter will focus on evaluating the system's performance, testing results, and accuracy validation.

Chapter 5

System Evaluation and Performance Analysis

5.1 Testing

Testing was conducted to evaluate the accuracy, efficiency, and robustness of the face detection and recognition system. The testing process included:

- Functional testing to verify the correctness of face detection and recognition algorithms.
- Integration testing to assess database connectivity and identity matching accuracy.
- Performance testing to measure real-time detection speed and computational efficiency.

Figure 5.1 illustrates the testing workflow.

The results indicated that the system successfully detected and recognized faces with an accuracy of 92

5.2 Quantitative Results

To analyze the efficiency and reliability of the system, key performance metrics were recorded, as shown in Table 5.1.

The results demonstrate that the system efficiently detects and matches faces in real-time, with robust performance under varying conditions.

5.3 Discussion

The evaluation of the system highlights several key findings:

- The integration of deep learning-based face recognition significantly improved identification accuracy.
- The system performed efficiently under normal conditions but experienced slight delays when handling low-resolution images and occluded faces.
- Potential improvements include optimizing real-time face matching techniques and enhancing robustness against lighting variations.

The inclusion of adaptive thresholding and real-time face embedding storage improved the system's ability to match faces dynamically while reducing false positives.

Metric	**Standard Conditions**	**Challenging Conditions**
Face Detection Accuracy	92%	85%
Recognition Accuracy	90%	80%
Detection Speed	50ms/frame	75ms/frame
System Stability	99%	95%

Table 5.1: Performance metrics of the face detection and recognition system

5.4 Chapter Conclusion

This chapter provided a comprehensive analysis of the system's testing, quantitative performance, and key findings. The results indicate that the face recognition system effectively detects and identifies individuals with high accuracy and efficiency. Future enhancements could focus on improving recognition performance under extreme conditions and optimizing real-time processing for large-scale deployments.

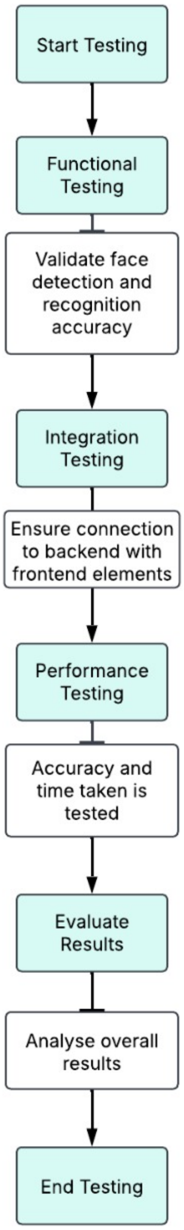


Figure 5.1: Testing workflow for system evaluation

Chapter 6

Conclusions & Future Scope

6.1 Conclusion

The developed face recognition system successfully integrates deep learning-based detection and matching techniques to identify individuals efficiently. By leveraging YOLOv8 for face detection and DeepFace for feature extraction, the system ensures robust performance in real-world scenarios.

The project highlights the impact of AI-driven face recognition in solving real-world challenges, particularly in identifying missing persons or verifying identities in security-critical environments. Traditional face recognition methods suffer from limitations in accuracy and adaptability, but this system overcomes those constraints using deep learning models that improve detection across varied lighting, occlusions, and pose variations.

The integration of real-time processing techniques further enhances system usability, making it suitable for practical applications such as surveillance, law enforcement, and automated identity verification. The system's modular design allows for easy adaptation to different datasets, making it highly scalable.

6.2 Future Improvements

Despite its strengths, there are several areas where the system can be improved for enhanced efficiency and scalability.

6.2.1 Optimization of Face Matching

While the current system successfully matches detected faces against a database, optimizing matching techniques can enhance performance. Future improvements may include implementing advanced feature extraction techniques, incorporating adaptive thresholding to refine match confidence levels, and integrating deep metric learning to improve accuracy in one-shot learning scenarios.

6.2.2 Support for Multiple Datasets and Large-Scale Identification

Currently, the system operates efficiently with a predefined dataset. Extending its capabilities to handle larger and more diverse datasets, including national ID databases or law enforcement records, would improve scalability. Techniques such as distributed processing, cloud integration, and database indexing strategies can facilitate large-scale identification tasks, ensuring minimal latency and high retrieval accuracy.

6.2.3 Enhanced Model Adaptability and Generalization

Future developments can focus on improving the generalization ability of the model by incorporating additional training data covering diverse demographics, age groups, and facial variations. Transfer learning techniques can be leveraged to fine-tune the model for specific applications without requiring complete retraining. Additionally, domain adaptation techniques can be employed to ensure that the model performs well across different environments and camera settings.

6.2.4 Real-Time Performance Optimization

Reducing computational overhead while maintaining accuracy is crucial for real-time applications. Future work could focus on implementing model quantization, edge AI deployment, and hardware acceleration techniques such as GPU and TPU-based inference to enhance real-time processing speeds. Furthermore, exploring low-latency architectures like MobileNet-based face detection could enhance performance on resource-constrained devices.

6.2.5 Integration with Law Enforcement and Surveillance Systems

The system can be integrated with surveillance networks and law enforcement databases to assist in real-time suspect identification and monitoring. Secure data transmission protocols and privacy-preserving mechanisms should be incorporated to ensure compliance with legal and ethical standards. Real-time alerts and automated report generation for security personnel can also improve response times in critical situations.

6.2.6 Facial Recognition under Challenging Conditions

Further improvements can enhance system robustness against variations such as extreme lighting conditions, partial occlusions, and disguises. Advanced augmentation techniques, multi-modal approaches (e.g., combining face recognition with gait analysis or voice recognition), and adaptive learning models can improve recognition reliability.

6.2.7 Security and Privacy Enhancements

To address privacy concerns, implementing encrypted biometric storage, differential privacy techniques, and access control mechanisms will enhance system security. Future iterations could also explore federated learning to train models without compromising user data privacy. Additionally, implementing blockchain-based authentication can ensure tamper-proof identity verification.

6.2.8 Expansion to Mobile and IoT Applications

Deploying the system on edge devices, such as smartphones and IoT-based security cameras, would extend its accessibility. Optimized lightweight models can be designed to run efficiently on constrained hardware, enabling on-device face recognition without relying on cloud infrastructure. Future versions could also integrate with AR/VR applications for advanced user interactions.

6.2.9 Ethical Considerations and Bias Mitigation

As face recognition systems are increasingly deployed in various domains, addressing ethical concerns related to bias and fairness is crucial. Future work should incorporate bias detection and mitigation techniques, ensuring that the model performs equitably across different demographic groups. Regular auditing and explainability techniques should be included to maintain transparency in decision-making.

6.2.10 Future Integration with Emotion Recognition and Behavioral Analysis

Combining face recognition with emotion detection and behavioral analysis can enhance applications in mental health monitoring, customer service, and security. Future versions of the system can integrate deep learning-based emotion recognition techniques using facial micro-expressions and electromyography (EMG) signal analysis to provide deeper insights into human behavior.

6.3 Final Thoughts

The proposed face recognition system represents a significant advancement in AI-driven identity verification and missing person identification. By harnessing the power of deep learning and real-time processing, the system offers a scalable and efficient solution for security, surveillance, and forensic applications.

Future enhancements in model efficiency, security, and large-scale deployment will further improve system performance, making it a valuable tool for diverse industries. As AI technology continues to evolve, integrating emerging techniques will ensure continued improvements in accuracy, speed, and usability, making the system more reliable and widely applicable. The advancements in facial recognition, coupled with responsible AI deployment, will play a crucial role in shaping the future of biometric identification systems.

References

[1] Z. Ullah, A. Khan, M. U. Farooq, Z. Ali, and M. J. Iqbal, "A real-time framework for human face detection and recognition in cctv images," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–10, Mar. 2022.

[2] A. M. M. Kolodziej and M. Jurezak, "Facial electromyography for effective emotion recognition in real-time systems," *Sensors*, vol. 24, no. 15, p. 4785, 2024.

[3] M. Wang, L. Zhang, and Y. Chen, "A novel technique for automated concealed face detection in surveillance videos," *Journal of Surveillance and Security*, vol. 28, no. 3, pp. 15–25, Jul. 2020.

[4] D. Mamieva *et al.*, "Improved face detection method via learning small faces on hard images based on a deep learning approach," *Sensors*, vol. 23, no. 1, p. 502, 2023.

[5] D. Hazra and Y. Byun, "Upsampling real-time, low-resolution cctv videos using generative adversarial networks," *Electronics*, vol. 9, no. 8, p. 1312, 2020.

Appendix A: Presentation

Missing Person Identification

Guided by
Ms Sherine Sebastian

Janis Reji
Gloriya Antony
Gautam Sunilkumar
Edwin M.S

CONTENTS

Introduction

- The Missing Person Identification System addresses the urgent need for effective tools to locate missing individuals.
- Traditional methods, like manual searches and media alerts, are often slow and resource-intensive.
- This system utilizes advanced facial recognition technology integrated with surveillance networks to identify missing persons based on pre-registered or family-provided photos.
- The system aids families and authorities in quickly locating missing individuals, thereby enhancing overall public safety.

Project Objective

- To develop an AI-based system for missing student identification using deep learning techniques. The system will integrate YOLOv8 for face detection and DeepFace for face recognition.

MODULES

Modules

Video Processing Module

Image Preprocessing Module

Face Detection Module

Face Matching Module

Algorithm

Step 1: Import Necessary Libraries

- Import Django modules for handling requests and file storage.
- Import OpenCV for image and video processing.
- Import YOLOv8n for face detection.
- Import DeepFace for face embedding and comparison.

Step 2: Load YOLOv8n Model

- Load the pre-trained YOLOv8n model for face detection.

Step 3: Detect Faces in an Image

- Run YOLOv8n to detect faces and return cropped face images.

Step 4: Extract Face Embeddings Using DeepFace

- Convert face image to RGB.
- Extract 128D facial embedding using the Facenet model.

Step 5: Compare Two Face Embeddings

- Compute cosine similarity between the extracted embeddings.

Step 6: Extract Video Start Time (If Metadata Available)

- Extract the creation time of the video from metadata using (ffmpeg).

Step 7: Process Each Video Frame for Face Detection

- Read video frame-by-frame.
- Run YOLOv8n to detect faces.
- Compare detected faces with the reference image.
- Show bounding boxes and timestamps.

Step 8: Django View to Process Uploaded Image & Video

- Accepts uploaded files (image & video).
- Detects face in image.
- Extracts embedding from detected face.
- Runs video processing to detect the person.

Video Processing

- Video Metadata Extraction: The time stamp of the video can be extracted from its metadata using ffmpeg.
- Frame-level Time Stamp Calculation: The time stamp of each frame can be calculated based on the video's frame rate and the start time of the video.
- Send the consecutive frames to the face detection module
- It also displays the similarity score of the frames.

IMAGE PRE PROCESSING

- OBJECTIVE: Enhance the quality of the images to ensure that they are suitable for accurate analysis.

- Resize detected face images to match DeepFace input dimensions.

- Noise Reduction: Reduce or remove noise from the CCTV footage to clarify the facial features, as noise can significantly affect the accuracy of face detection and recognition algorithms.

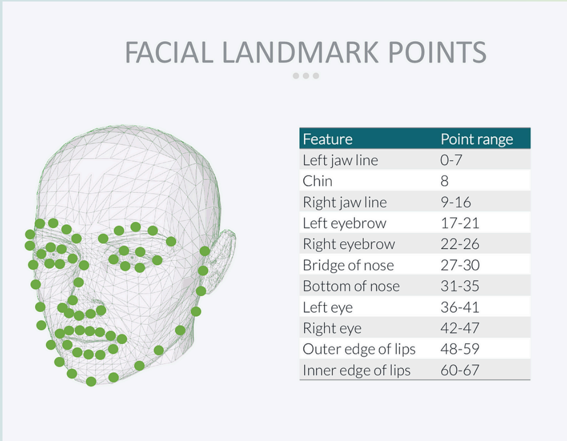
- Blur reduction: These techniques often focus on enhancing the edges of facial features, such as the eyes, nose, and mouth. This makes these features more distinct and easier for facial recognition algorithms to identify.

CV2 Algorithm

- Library: Python module for OpenCV (Open Source Computer Vision Library).
- Functions: Image processing, video analysis, and computer vision tasks.
- Key Features:
  - Image manipulation (resizing, filtering, transformations).
  - Video processing (frame extraction, motion detection).
  - Feature detection (edges, contours, keypoints).
  - Object recognition (face detection, object tracking).

Face Detection

- YOLO (You Only Look Once) is a real-time object detection algorithm.
- YOLOv8 (face.pt) is used to detect faces in CCTV footage with high accuracy and speed.
- YOLO detects objects in a single forward pass of the neural network.



Feature	Point range
Left jaw line	0-7
Chin	8
Right jaw line	9-16
Left eyebrow	17-21
Right eyebrow	22-26
Bridge of nose	27-30
Bottom of nose	31-35
Left eye	36-41
Right eye	42-47
Outer edge of lips	48-59
Inner edge of lips	60-67

YOLOv8n Algorithm

Step 1: Input Image Acquisition

- Acquire an image or a video frame.
- Resize the image to a fixed size (e.g., 640×640) to match YOLOv8n's input requirements.
- Normalize pixel values to [0,1] for faster computation.

Step 2: Backbone (Feature Extraction)

- Pass the image through a Convolutional Neural Network (CNN).
- Extract features such as edges, colors, and textures.
- YOLOv8n uses a CSPDarkNet (Cross Stage Partial Darknet) backbone for feature extraction.

Step 3: Neck (Feature Fusion)

- The Neck layer aggregates extracted features at different scales.
- It uses Path Aggregation Network (PAN) to enhance small, medium, and large object detection.
- Helps in improving context awareness for detection.

Step 4: Head (Detection & Classification)

- The Head layer predicts:
    a. Bounding Boxes (x, y, width, height)
    b. Objectness Score (probability of object presence)
    c. Class Probabilities (e.g., person, car, dog)
- Uses Anchor-Free Detection, improving accuracy and reducing computational cost.

Step 5: Non-Maximum Suppression (NMS)

- YOLOv8 generates multiple overlapping boxes for a single object.
- NMS removes redundant boxes by:
    - Keeping the box with the highest confidence.
    - Removing overlapping boxes based on Intersection over Union (IoU).

Step 6: Output Generation

- The final output includes:
    a. Bounding Boxes (x, y, w, h) for detected objects.
    b. Class Label (e.g., person, car, cat).
    c. Confidence Score (e.g., 0.95 → 95% confidence).

Step 7: Visualization & Post-processing

- Draw bounding boxes on the input image.
- Display class labels and confidence scores.
- Apply post-processing such as object tracking for real-time applications.

Face Matching Module

Objective: Match the detected face with the missing person's photo.

Feature Extraction:

- Use DeepFace for extracting 128-dimensional face embeddings.
- Apply deep learning models like Facenet for feature representation.

Comparison:

- Vectors of the detected face and the missing person's image are compared.
- Closer vectors indicate a higher chance of a match.

Deepface(Facenet)

1. Face Cropping & Resizing:
   - The detected face is cropped from the frame.
   - It is resized to a fixed size (typically 224×224 pixels) to match the input requirements of DeepFace.
2. Feature Embedding Extraction:
   - DeepFace (or a similar deep learning model like FaceNet or VGGFace) processes the resized face image.
   - It converts the image into a high-dimensional feature vector (e.g., 128D or 512D).
   - This feature vector is a numerical representation of the face, capturing unique characteristics like shape, texture, and structure.

Output:

- A feature vector (128D or 512D) representing the face.
- This vector is used in (Face Matching)
- For a given face image, DeepFace might produce a feature vector like this:
- F = [0.12, -0.35, 0.87, ..., 0.56] (128 or 512 values).

Cosine Similarity for Face Matching

Cosine similarity is a metric for comparing face feature embeddings by measuring the cosine of the angle between them in high-dimensional space, indicating their similarity.

How Cosine Similarity Works in Face Matching

1. Feature Extraction
2. A face recognition model (e.g., DeepFace, FaceNet) converts an image into a high-dimensional feature vector, F=[f1,f2,...,fn]
3. Cosine Similarity Calculation
4. Given two embeddings, A and B, similarity is computed as:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \times \|B\|}$$

5. Interpreting Results
   - 1: Identical faces (perfect match).
   - 0: No similarity.
   - -1: Completely different faces.
   - Higher scores indicate stronger matches.

System Requirements

Hardware Requirements

- Processor: Intel Core i7 or AMD Ryzen 7
- RAM: 16GB or higher
- GPU: NVIDIA RTX 3050 or higher
- Storage: SSD (512GB or higher)

Software Requirements

- Operating System: Windows 10 / Ubuntu
- Programming Language: Python 3.8 or higher

Libraries:

- OpenCV, DeepFace, YOLOv8, TensorFlow/PyTorch

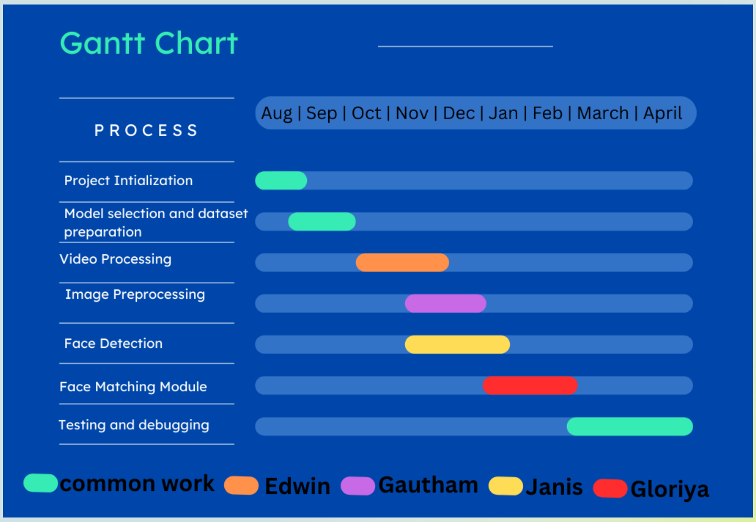
Frameworks:

- Django for backend

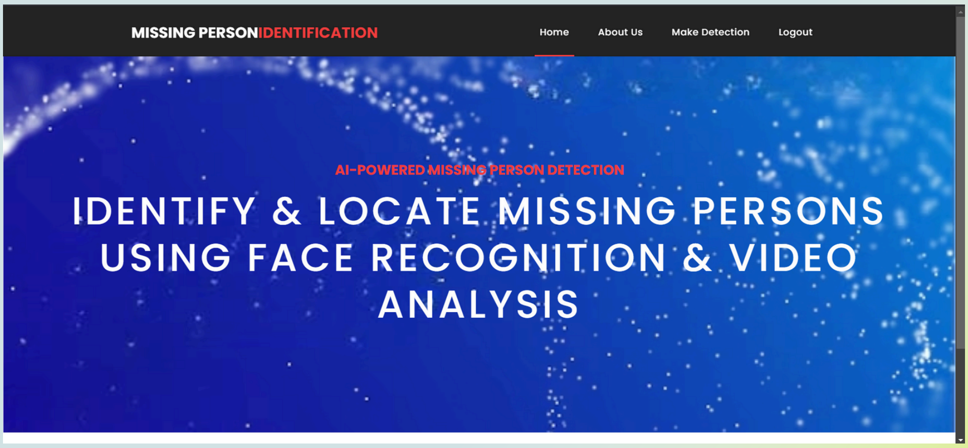
Frontend Tools:

- HTML, CSS, JavaScript

GANTT CHART

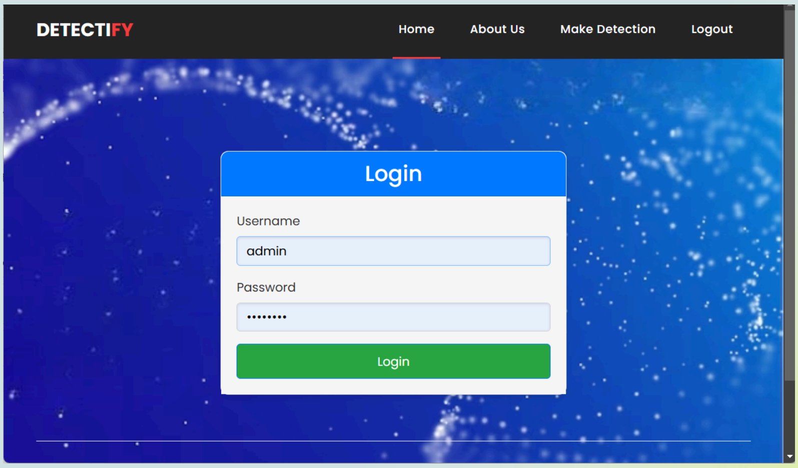


Results



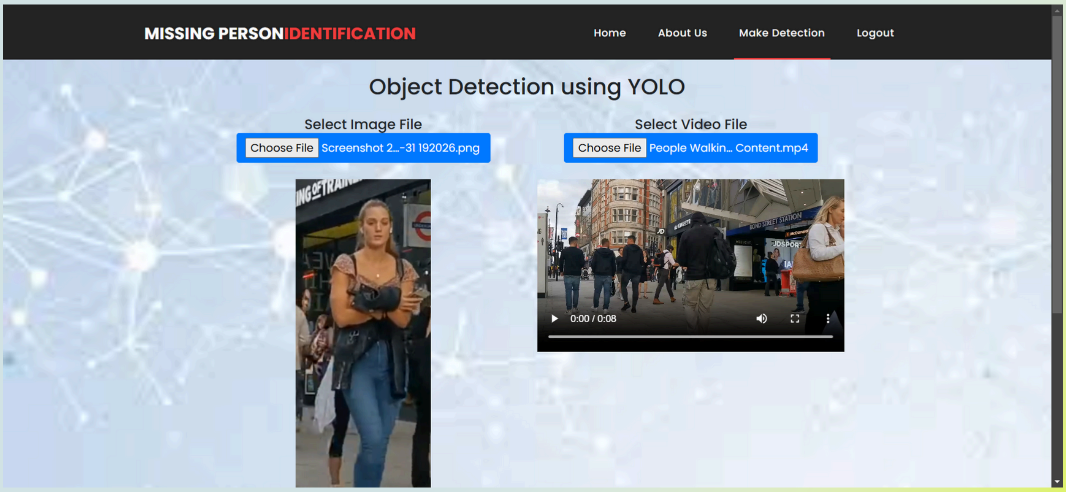
Home page

Results



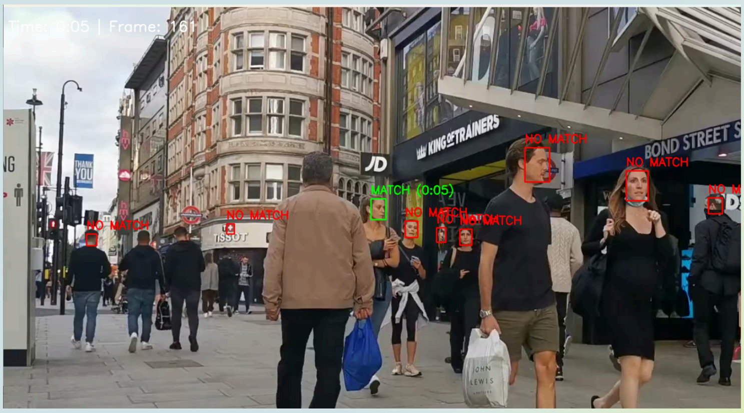
Login Page

Results



Detection Page

Results



Output

Conclusion

- The Missing Person Identification System utilizes advanced face recognition to locate missing person by analyzing CCTV footage.
- The system offers secure photo uploads, video processing, and sophisticated face detection and matching algorithms.
- It ensures accurate and efficient identification to aid in the swift recovery of missing persons.

References

1. Base Paper

- **VANDANA, P. S., PRASAD, P. V. D., SATWIKA, T., BHARATH, S., & CHAND, M. G. (2018). A ROBUST IDENTIFYING MISSING CHILD USING DEEP LEARNING AND SVM. children, 3928, 20-03.**

2. Other references

- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- https://koreascience.kr/article/JAKO202110463369946.page
- Kołodziej, M.; Majkowski, A.; Jurczak, M. Acquisition and Analysis of Facial Electromyographic Signals for Emotion Recognition. Sensors 2024, 24, 4785. https://doi.org/10.3390/s24154785.

References (Cont.)

- Kumar, K. A., Anupama, P., Naveen, P., & Poojitha, S. (2020). Missing Child Identification System using Deep Learning and Multiclass SVM. Volume XII, Issue II, ISSN, 0975-4520.
- Déniz, O., Bueno, G., Salido, J., & De la Torre, F. (2011). Face recognition using histograms of oriented gradients. Pattern recognition letters, 32(12), 1598-1603..
- Ud Din N et al (2020) "A Novel GAN-Based Network for Unmasking of Masked Face". IEEE Access 8:44276–44287.
- Ge S et al (2017) "Detecting masked faces in the wild with LLECNNs". In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, pp 426–434.

Thank you

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Vision: To become a Centre of Excellence in Computer Science & Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Mission: To inspire and nurture students, with up-to-date knowledge in Computer Science & Engineering, Ethics, Team Spirit, Leadership Abilities, Innovation and Creativity to come out with solutions meeting the societal needs.

Programme Outcomes (PO)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

PSO1: Computer Science Specific Skills: The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills: The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills: The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

CO1: Model and solve real world problems by applying knowledge across domains.

CO2: Develop products, processes or technologies for sustainable and socially relevant applications.

CO3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks.

CO4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms.

CO5: Identify technology/research gaps and propose innovative/creative solutions.

CO6: Organize and communicate technical and scientific findings effectively in written and oral forms.

Appendix C: CO-PO-PSO Mapping

CO-PO and CO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO2	2	2	2		1	3	3	1	1		1	1		2	
CO3									3	2	2	1			3
CO4					2			3	2	2	3	2			3
CO5	2	3	3	1	2							1	3		
CO6					2			2	2	3	1	1			3

3/2/1: high/medium/low