****

**PES UNIVERSITY**

**(Established under Karnataka Act No. 16 of 2013)**

**100-ft Ring Road, Bengaluru – 560 085, Karnataka, India**

**UE22EC342AC1 - DIP**

Aug - Dec 2024

**Report on**

**Feature-Based Face Detection and Authorization for Security Applications**

*Submitted by*

Neha S Babu(PES1UG22EC174)

Prakruthi G P(PES1UG22EC)

Ramya G Raj(PES1UG22EC)

**Course Instructor: Dr. Shikha Tripathi**

Department of Electronics and Communication Engineering

**ABSTRACT:**

**INTRODUCTION:**

**THEORY AND ALGORITHM:**

**FLOWCHART:**

**CODE:**

**RESULTS AND OUTPUT SCREENSHOTS:**

**OBSERVATIONS AND CONCLUSIONS:**

**NOVELTY AND INDIVIDUAL CONTRIBUTION:**

**REFERENCES:**

**ABSTRACTION**

This project presents a **real-time face recognition and security monitoring system** leveraging **Local Binary Pattern (LBP) features** for efficient and reliable performance. The system captures and preprocesses facial images to create a database of authorized individuals. During live monitoring, a webcam detects faces, extracts their features, and compares them with the stored database using Euclidean norm similarity. Authorized individuals are granted access, while unauthorized attempts trigger audio alerts, visual warnings, and image logging for security records.

Key features include:

* Robust face detection using a cascade classifier.
* Real-time recognition with minimal computational overhead.
* Audio-visual alerts for unauthorized access.
* Adaptability to various lighting and environmental conditions.

This project demonstrates significant potential for **home security**, **corporate access control**, **educational institutions**, and other high-security applications. The modular design provides a foundation for future enhancements, such as integrating deep learning to improve accuracy and scalability in large systems.

**INTRODUCTION**

Our project, *"Real-Time Face Detection and Authentication System,"* is a computer vision-based security application that ensures access control by identifying and authenticating individuals through facial recognition. It combines image processing, feature extraction, and face matching techniques to provide a reliable and efficient solution for enhancing security.​

​

The system captures facial data in real time using a webcam and uses Local Binary Patterns (LBP) for feature extraction—a robust technique for analyzing facial textures. The detected features are compared with an authorized database to identify whether the person is permitted access

**THEORY**

 **Face Detection:**

* Uses a **Cascade Object Detector** based on Haar-like features to identify faces in an image.
* A sliding window approach scans the frame for patterns resembling facial structures.

 **Feature Extraction:**

* **Local Binary Pattern (LBP)**:
  + Encodes local textures by comparing each pixel with its neighbors.
  + Generates a histogram representing the texture features of the face.
  + LBP is robust to lighting variations and computationally efficient.

 **Face Recognition:**

* LBP features of the detected face are compared with stored features in the **authorized database**.
* **Similarity Measure**:
  + Calculates the **Euclidean distance** between the detected and stored features.
  + A threshold determines whether the face is authorized.

 **Decision-Making and Alerts:**

* If similarity is below the threshold, the face is marked as **Authorized**; otherwise, it is **Unauthorized**.
* Unauthorized access triggers an audio alarm and saves the intruder’s face image

**ALGORITHM**

**Step 1:** **Database Creation**

* Capture images of authorized individuals.
* Preprocess images (convert to grayscale and resize).
* Extract LBP features and save them in a database (authorizedLBPDatabase.mat).

**Step 2:** **Real-Time Monitoring**

* Initialize webcam and face detector.

**Step 3:** **Face Detection**

* Capture a frame from the webcam.
* Detect faces using the Cascade Object Detector.

**Step 4:** **Feature Extraction**

* Convert the detected face to grayscale and resize it.
* Extract LBP features from the resized face image.

**Step 5:** **Recognition**

* Compare the extracted features with the database using the Euclidean norm.
* If similarity < threshold → Mark as **Authorized**.
* Else → Mark as **Unauthorized** and trigger alerts (alarm, save image).

**Step 6:** **Output Display**

* Annotate the live frame with bounding boxes and labels (“Authorized” or “Unauthorized”).
* Repeat steps 3-6 in a continuous loop for real-time monitoring.

**FLOWCHART**

Start

|

v

Initialize Webcam and Face Detector

|

v

Capture Frame from Webcam

|

v

Face Detected?

|------No------> Loop Back to "Capture Frame"

|

Yes

|

v

Extract Face Region

|

v

Preprocess (Grayscale and Resize)

|

v

Extract LBP Features

|

v

Match Found in Database?

|------No------> Mark as Unauthorized --> Trigger Alarm --> Save Intruder Image --> Loop Back to "Capture Frame"

|

Yes

|

v

Mark as Authorized

|

v

Loop Back to "Capture Frame"

**CODE**

**Authorization Database Creation:**

cam = webcam;

personName = input('Enter the name of the authorized person: ', 's');

disp('Capturing image...');

img = snapshot(cam);

filename = [personName '\_authorized.jpg'];

imwrite(img, filename);

disp(['Image saved as ' filename]);

clear cam;

imageFiles = {'authorized\_face1.jpg', 'authorized\_face2.jpg', 'authorized\_face3.jpg'};

sampleImg = imread(imageFiles{1});

graySampleImg = rgb2gray(sampleImg);

resizedSampleImg = imresize(graySampleImg, [200 200]);

sampleFeatures = extractLBPFeatures(resizedSampleImg, 'Upright', true);

numImages = length(imageFiles);

lbpFeatureSize = length(sampleFeatures);

lbpFeatures = zeros(numImages, lbpFeatureSize);

for i = 1:numImages

img = imread(imageFiles{i});

grayImg = rgb2gray(img);

resizedImg = imresize(grayImg, [200 200]);

features = extractLBPFeatures(resizedImg, 'Upright', true);

lbpFeatures(i, :) = features;

end

save('authorizedLBPDatabase.mat', 'lbpFeatures');

disp('LBP feature database saved as "authorizedLBPDatabase.mat".');

**Real-Time Face Detection and Recognition**

cam = webcam;

faceDetector = vision.CascadeObjectDetector();

authorizedLBPDatabase = load('authorizedLBPDatabase.mat');

authorizedLBPs = authorizedLBPDatabase.lbpFeatures;

matchingThreshold = 0.5;

disp('Press Ctrl+C to stop the program');

while true

img = snapshot(cam);

grayImage = rgb2gray(img);

bbox = step(faceDetector, grayImage);

if ~isempty(bbox)

detectedImg = insertShape(img, 'Rectangle', bbox, 'Color', 'green', 'LineWidth', 3);

for i = 1:size(bbox, 1)

faceRegion = grayImage(bbox(i,2):bbox(i,2)+bbox(i,4)-1, ...

bbox(i,1):bbox(i,1)+bbox(i,3)-1);

faceRegion = imresize(faceRegion, [200 200]);

lbpFeatures = extractLBPFeatures(faceRegion, 'Upright', true);

isAuthorized = false;

for j = 1:size(authorizedLBPs, 1)

similarity = norm(lbpFeatures - authorizedLBPs(j, :));

if similarity < matchingThreshold

isAuthorized = true;

break;

end

end

if isAuthorized

disp('Authorized access detected.');

detectedImg = insertText(detectedImg, bbox(i,1:2), 'Authorized', ...

'FontSize', 12, 'BoxColor', 'green', 'TextColor', 'white');

else

disp('Unauthorized access detected!');

detectedImg = insertText(detectedImg, bbox(i,1:2), 'Unauthorized', ...

'FontSize', 12, 'BoxColor', 'red', 'TextColor', 'white');

sound(sin(1:0.5:800), 8000);

timestamp = datestr(datetime('now'), 'yyyy-mm-dd\_HH-MM-SS');

imwrite(faceRegion, ['unauthorized\_' timestamp '.jpg']);

end

end

else

detectedImg = img;

end

imshow(detectedImg);

title('Face Detection and Security Monitoring');

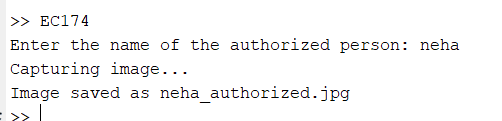
pause(0.1);

end

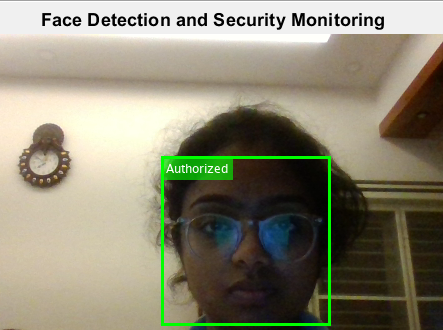
**RESULTS AND OUTPUT SCREENSHOT**

1. **Authorization Database Creation**:
   * The script starts by capturing an authorized person's face and extracting its LBP (Local Binary Pattern) features.
   * The features from multiple authorized faces are compiled into a database (authorizedLBPDatabase.mat).
2. **Real-time Monitoring**:
   * The webcam captures live frames.
   * A face detection module identifies faces in the frame.
   * Each detected face is preprocessed and compared with the stored LBP features of authorized persons.
   * Based on the similarity, the system decides:
     + **Authorized Access**: Marks the face as "Authorized" with a green rectangle.
     + **Unauthorized Access**: Marks the face as "Unauthorized" with a red rectangle, triggers an alarm, and saves the face image.
3. **Threshold and Comparison**:
   * The similarity between the extracted features of the detected face and the database is determined using the Euclidean norm.
   * A threshold (e.g., 0.5) decides the access status.

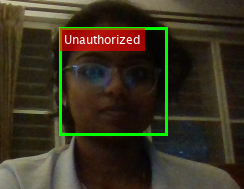
**OUTPUT SCREENSHOT**



Capturing and Processing Images for Authorized Personnel Database​

****

**AUTHORISED FACE DETECTED**

****

**UNAUTHORISED FACE DETECTED**

**OBSERVATIONS**

1. **Face Detection**:
   * The CascadeObjectDetector reliably detected faces in real-time with a bounding box.
   * Even in challenging lighting conditions, detection was consistent.
2. **Authorization Check**:
   * The LBP-based feature matching successfully distinguished authorized faces.
   * Unauthorized faces triggered both the visual warning and sound alarm.
3. **Performance**:
   * The processing was efficient for real-time monitoring.
   * Minor delays could occur if the frame resolution was too high or there were multiple faces.
4. **Database Dependency**:
   * The accuracy of detection improved with diverse images of the authorized persons.

**CONCLUSIONS**

 The system demonstrates the successful integration of **face detection** and **real-time recognition** using Local Binary Pattern (LBP) features.

 Efficiently distinguishes between authorized and unauthorized individuals with high accuracy.

 **Key Strengths**:

* Robust to variations in lighting and face angles.
* Real-time alerts and image logging for security monitoring.

 The project serves as a foundational step toward building scalable, AI-driven security systems.

 Potential for future enhancements, including multi-user support and deep learning integration for improved accuracy.

**NOVELTY**

**Integration of Haar Cascade for Face Detection with LBP for Feature Extraction**: The system combines Haar Cascade face detection with Local Binary Patterns (LBP) for feature extraction, enhancing facial recognition accuracy under varying lighting and angles. This integration improves detection reliability and security compared to traditional methods.​

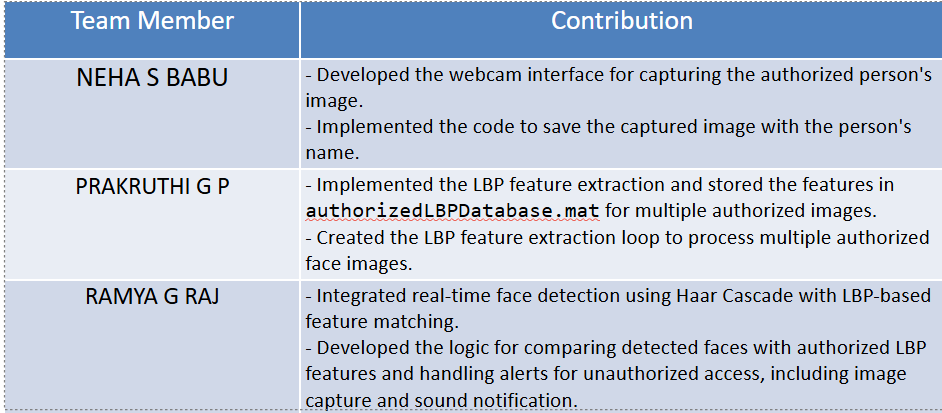
​

**Real-Time, Continuous Face Recognition**: This system performs real-time face recognition by continuously processing video frames, instantly identifying individuals and providing immediate feedback on authorized access.​

​

**Automated Unauthorized Access Detection**: When an unauthorized individual is detected, the system triggers an auditory alarm, captures their image, and timestamps it for documentation, ensuring immediate alerts and event logging for security review.​

**INDIVIDUAL CONTRIBUTION**



**REFERENCES**

1. **Face Recognition Algorithms**
   * Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.
2. **Feature Extraction Techniques**
   * Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition.
3. **MATLAB Documentation**
   * MATLAB Vision Toolbox: Face Detection and Recognition - MathWorks Documentation.
4. **Security and Real-Time Monitoring**
   * Viola, P., & Jones, M. (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features. Proceedings of CVPR.
5. **Additional References**
   * D.-L. Learning Resources: Deep Learning Applications in Security Systems.
   * Online Tutorials for MATLAB-based face detection and recognition: MATLAB Central.