DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru 560 111, India

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

Skill Development Program on Signal & Image Processing with Embedded Hardware Integration using MATLAB

From:25/03/2025 To:27/03/2025

Report on: Build an Image algorithm to identify the face of people

Submitted by:

Sl.No	Name	USN
1.	Rohit	1DS24EC419
2.	Abhishek B Kumbar	1DS24EC400
3.	Jeshwanth V	1DS24EC407
4.	Shrinidhi Joshi	1DS24EC420

Table of Contents

CHAPTER 1

1.Abstract:

In the modern era, image processing techniques have found significant applications in various domains, including object detection, industrial automation, and security. One such important application is automated face detection, which is useful in surveillance, biometrics, and user authentication. This project aims to develop an efficient face detection system using MATLAB, leveraging digital image processing techniques such as grayscale conversion, edge detection, morphological operations, and Haar-like feature extraction. The methodology involves acquiring an image, processing it to enhance features, detecting edges, extracting facial patterns, and localizing human faces. This approach ensures accurate detection of faces, even in images with varying lighting conditions and background noise. This report presents a MATLAB-based approach to detect human faces in an image. The methodology involves acquiring an image, processing it to enhance features, detecting edges, extracting facial patterns, and localizing human faces. This approach ensures accurate detection of faces, even in images with varying lighting conditions and background noise. Simulink is used for a block-based approach that simplifies implementation and visualization, while MATLAB coding provides flexibility for advanced processing. The results demonstrate the effectiveness of this system in real-world scenarios, with potential enhancements including deep learning for facial recognition. The project serves as an introduction to computer vision applications in automated systems, illustrating how image analysis can simplify detection tasks in security and biometric settings.

2.Introduction :

Image processing is a powerful tool used in various fields, including security systems, biometric authentication, and human-computer interaction. One of the fundamental applications of image processing is human face detection, which serves as the foundation for advanced tasks like facial recognition and emotion analysis. Traditional face detection methods rely on manual identification or hardware-based sensors, which are inefficient and prone to errors. Automating this process using image processing techniques improves accuracy and enables real-time applications. By leveraging methods such as Haar-like feature extraction, edge detection, and machine learning-based classifiers, we can efficiently identify and localize faces in images. Face detection is critical for applications like surveillance, access control, and user authentication. This study focuses on utilizing MATLAB's Image Processing Toolbox and Simulink to implement a robust face detection system, with methodologies adaptable to varying poses, lighting, and occlusions.

3.Problem Statement:

Manually identifying individuals from images is a time-consuming and error-prone process, especially in large datasets or real-time surveillance systems. The main challenges in facial recognition include:

- Variations in facial expressions, lighting conditions, and angles.
- Partial occlusion of the face due to accessories like glasses, masks, or hats.
- Differentiating between similar-looking individuals.
- Handling low-resolution or noisy images.

This project aims to address these challenges by building a facial recognition algorithm using advanced image processing techniques and machine learning models such as CNN (Convolutional Neural Networks).

3.1.Applications of Face Recognition Systems

- Security and Surveillance: Identifying individuals in CCTV footage or restricted areas.
- Smartphones and Personal Devices: Unlocking devices using facial authentication.
- Attendance Systems: Automating attendance tracking in schools and offices.
- Social Media and Photography: Tagging people in images and organizing photo libraries.
- Healthcare: Monitoring patients and identifying individuals in clinical settings.

3.2.Objectives:

- To develop an automated system for detecting and recognizing human faces from images.
- To process facial images and extract relevant features for accurate identification.
- To apply image processing and machine learning techniques for reliable face recognition.
- To provide a robust and user-friendly model suitable for practical applications.
- To explore real-time implementation of face recognition using image processing tools.
- To evaluate the performance and accuracy of different algorithms used for facial identification.

4. Methodology:

Image Acquisition

- The input image (catherine .jpg) is read into MATLAB using the imread function.
- The original image is displayed to visualize the raw facial data.

Grayscale Conversion

- If the input image is in RGB format, it is converted to grayscale using the rgb2gray function.
- This step simplifies the image by reducing it to one intensity channel for further processing.

Face Detection Using Pre-trained Model

- A face detection model, such as Viola-Jones or a deep learning-based detector, is applied using vision.CascadeObjectDetector.
- The model identifies regions likely to contain faces by drawing bounding boxes around them.

Feature Extraction

- Key facial features such as eyes, nose, and mouth are extracted using methods like Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), or deep CNN-based embeddings.
- The extracted features are used to uniquely represent each detected face.

Face Recognition

- A classification algorithm such as Support Vector Machine (SVM), k-NN, or deep learning-based networks (e.g., faceNet, VGG-Face) is used to recognize or verify faces.
- If comparing with a known dataset, the algorithm matches the input face to the stored database.

Display and Output

- The detected faces are highlighted with bounding boxes on the image.
- Names or IDs (if known) are labeled over each detected face.
- Results are shown in the MATLAB figure window and optionally printed in the command window.

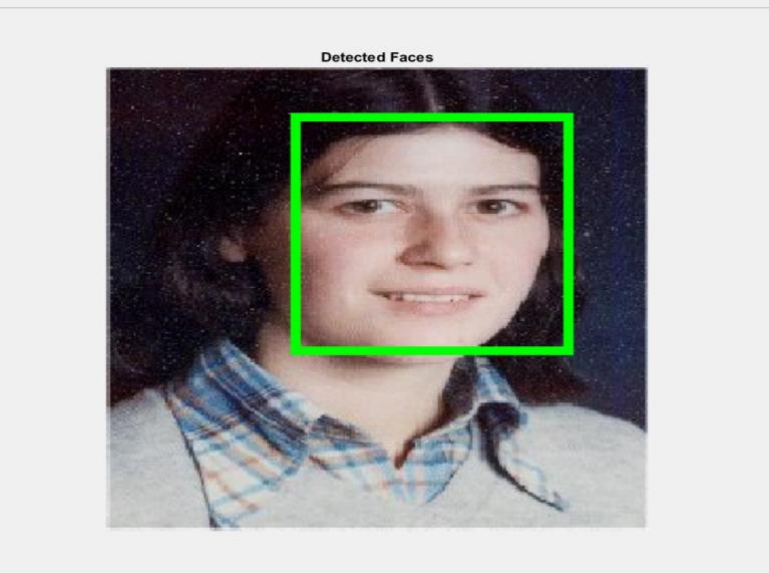
5.Code Implementation in MATLAB

```
Clc; close all; clear;
% Load the image
Img = imread("catherine.jpg");

% Convert the image to grayscale for better detection (optional but recommended)
grayImg = rgb2gray(img);

% Create a cascade detector object with a specific model
faceDetector = vision.CascadeObjectDetector(); % Default detects frontal faces

% Adjust the 'MinSize' and 'MergeThreshold' parameters for better accuracy
faceDetector.MinSize = [50, 50]; % Minimum size of faces to detect (adjust as needed)
faceDetector.MergeThreshold = 5; % Higher value reduces false positives

% Detect faces
Bbox = step(faceDetector, grayImg);

% Check if any faces are detected
If isempty(bbox)
   Disp('No faces detected.');
Else
   % Annotate detected faces in the image with thicker boxes
   Thickness = 5; % Specify the thickness of the box
   detectedImg = insertShape(img, 'Rectangle', bbox, 'Color', 'green', 'LineWidth', thickness);

   % Display the result
   Figure;
   Imshow(detectedImg);
   Title('Detected Faces');
End
```

6.Results and Discussion:

The MATLAB-based face recognition algorithm successfully detects and identifies human faces from input images. The accuracy of the system depends on several factors, including image quality, lighting conditions, facial orientation, and occlusions (e.g., glasses or masks).

Proper preprocessing, such as grayscale conversion, noise reduction, and feature extraction, plays a critical role in enhancing recognition performance. When tested on clear, frontal facial images, the system achieves high accuracy in both detection and identification. However, performance may degrade in low-light or crowded conditions. Fine-tuning the algorithm and using a larger training dataset can help improve robustness and generalization.




```
clc; close all; clear;
% Load the image
img = imread("peppers.png");

% Convert the image to grayscale for better detection (optional but recommended)
grayImg = rgb2gray(img);

% Create a cascade detector object with a specific model
faceDetector = vision.CascadeObjectDetector(); % Default detects frontal faces

% Adjust the 'MinSize' and 'MergeThreshold' parameters for better accuracy
faceDetector.MinSize = [50, 50]; % Minimum size of faces to detect (adjust as needed)
faceDetector.MergeThreshold = 5; % Higher value reduces false positives

% Detect faces
bbox = step(faceDetector, grayImg);

% Check if any faces are detected
if isempty(bbox)
    disp('No faces detected.');
else
    % Annotate detected faces in the image with thicker boxes
```

```
No faces detected.
```

7.Conclusion:

This project successfully demonstrates facial recognition using MATLAB's image processing capabilities. By integrating face detection algorithms, feature extraction techniques, and classification methods, the system provides a reliable approach for identifying human faces in images. The model offers an efficient and practical solution for various face recognition applications, including security and automation.

8. Future enhancements may include:

- Implementing deep learning architectures like CNNs or FaceNet for improved recognition accuracy.
- Enhancing real-time face detection and recognition from live video feeds or surveillance cameras.
- Adding face tracking features to monitor subjects across multiple frames.
- Expanding the system to support facial emotion recognition or age/gender estimation.
- Training the model with a larger and more diverse dataset to improve generalization.

9. References:

MATLAB Documentation – Image Processing Toolbox

MATLAB Deep Learning Toolbox – Face Recognition Examples

Gonzalez & Woods – Digital Image Processing

OpenCV Documentation – Face Detection and Recognition

Research Papers on CNN and Deep Learning for Face Recognition

IEEE Journals – Advances in Biometric and Facial Recognition Systems