Face Recognition using Local binary pattern Algorithm

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR

THE AWARD OF DEGREE

OF

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted by:

SAKET KUMAR

AGARWAL

2K21/ISY/21

Under the supervision

of

Prof. Kapil Sharma

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2023

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

STUDENT DECLARATION

I, Saket Kumar Agarwal of Information Systems department, hereby declare that the thesis titled "**Face Recognition using Local binary pattern Algorithm**" is submitted by me in partial fulfilment of the requirement for the award of degree inInformation Systems (Masters) is original and not copied from any source without citation. This work has not previously formed the basis for the award of and Degree, Diploma Associate ship, Fellowship or any other kind of similarity in title or recognition.

.

Date: SAKET KUMAR AGARWAL
Place: 2K21/ISY/21

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

This is to certify that Mr. SAKET KUMAR AGARWAL (2K21/ISY/21) has finished the major project titled "**Face Recognition using Local binary pattern Algorithm"** as a partial fulfillment for the award of Master of Technology degree in Information Systems by Delhi Technological University. Under my supervision and guidance during the academic session 2021-2023.

Date:
Place:

Prof. Kapil Sharma
SUPERVISOR
Professor
Department of Information Technology
Delhi Technological University

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I want to thank my major project guide **Prof. Kapil Sharma**, lT Dept., Delhi Technological University, for his kind support and guidance that he has provided me in order to finish this major project. I would also like to take this opportunity to thank the institute for providing me with the adequate infrastructure .At the end I want to thank my family and friends for their support.

Date: **SAKET KUMAR AGARWAL**
Place: **2K21/ISY/21**

ABSTRACT

Face recognition is a prominent field in computer vision that aims to automatically identify individuals based on their facial characteristics. In this study, we propose a face recognition system utilizing the Local Binary Pattern (LBP) algorithm. LBP is a texture descriptor that characterizes the local spatial patterns within an image, which can effectively capture the discriminative features of facial textures. The proposed face recognition system consists of two main stages: training and testing. In the training stage, a database of face images is utilized to extract LBP features from facial regions of interest. These features are then used to train a classification model, such as a support vector machine (SVM) or a neural network, to learn the discriminative patterns associated with each individual in the database.

During the testing stage, the LBP algorithm is applied to extract features from an input face image. These features are compared with the learned patterns in the classification model to determine the identity of the individual. The proposed system provides robustness against variations in illumination, facial expressions, and pose, making it suitable for real-world face recognition applications. To evaluate the performance of the proposed system, extensive experiments were conducted on benchmark face databases, including LFW, CASIA-Web Face, and FRGC. The results demonstrate that the LBP-based face recognition system achieves competitive accuracy rates compared to state-of-the-art methods, while maintaining computational efficiency.

This study presents a face recognition system based on the Local Binary Pattern algorithm, which offers a robust and efficient solution for identifying individuals based on their facial characteristics. The proposed system holds great potential for various applications, including security systems, access control, and surveillance technologies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Problem Definition

LBPH stands for Local Binary Patterns Histograms. It is a feature extraction algorithm commonly used in computer vision and image processing for texture analysis and facial recognition tasks. LBPH operates by dividing an image into small regions and extracting local binary patterns (LBP) from each region. The Local Binary Pattern is a simple yet powerful texture descriptor that encodes the relationship between a pixel and its neighbors in a binary code. It characterizes the local structure and texture information within an image region. By comparing these binary patterns, LBPH can capture the texture patterns present in different regions of an image. LBPH computes a histogram of the local binary patterns extracted from each region, representing the frequency distribution of the different pattern types [1]. The histogram provides a compact representation of the texture information contained in the image, which can be used for various tasks such as object recognition, image retrieval, and facial recognition.

1.2 Objective

The objective of the LBPH (Local Binary Patterns Histograms) algorithm is to extract local texture information from an image or region, represent it through histograms of local binary patterns, and enable efficient pattern matching and recognition tasks. LBPH aims to capture and describe the texture patterns by encoding relationships between pixels and their neighbors, while being robust to illumination variations. By extracting discriminative features and representing them in a histogram format, LBPH provides a compact and informative representation of texture, facilitating tasks such as object recognition, image retrieval, and facial recognition [2].

1.3 Overview

LBPH (Local Binary Patterns Histograms) is an algorithm used for texture analysis and pattern recognition tasks in computer vision. It aims to capture and represent local texture information within an image or a region of interest. The algorithm achieves this by dividing the image into small regions and extracting local binary patterns (LBP) from each region. These binary patterns encode the relationships between a pixel and its neighboring pixels, capturing the local texture variations and structures.

After extracting the local binary patterns, LBPH computes histograms to represent the frequency distribution of different pattern types within each region. These histograms provide a concise and informative representation of the texture features present in the image. By comparing these histograms, LBPH enables efficient pattern matching and recognition tasks [3].

One of the advantages of LBPH is its robustness to illumination variations. It achieves this by considering the local differences in pixel values rather than absolute values, making it less sensitive to changes in lighting conditions.

LBPH has found widespread applications in various fields, including object recognition, image retrieval, and particularly in facial recognition systems. In facial recognition, LBPH can capture the local texture patterns that are essential for distinguishing different individuals.

Overall, LBPH provides a simple yet effective approach to extract and represent local texture information, making it a valuable tool for tasks that require texture analysis and pattern recognition in computer vision [4].

CHAPTER 2

LITERATURE REVIEW

2.1 What is Face Recognition?

Face recognition, also known as facial recognition, is a technology that enables the identification or verification of individuals by analyzing and comparing their facial features (see [11]). It is a biometric method that uses computational algorithms to analyze patterns, shapes, and unique characteristics of a person's face [3].

The process of face recognition involves several steps:

Face detection: The system identifies and locates human faces in images or video frames. It uses techniques such as machine learning and computer vision to detect facial regions accurately.

Face alignment: Detected faces are then aligned to a standardized position and orientation. This step helps normalize the facial features and improve accuracy during subsequent analysis.

Feature extraction: Specific facial features, such as the distance between the eyes, shape of the nose, and other distinguishing characteristics, are analyzed. These features are converted into a mathematical representation or feature vector, which serves as a unique identifier for each individual.

Face matching: The extracted facial features or feature vectors are compared against a database of known faces or templates. The system uses algorithms to calculate the similarity or dissimilarity between the features of the detected face and the features of the faces in the database [8].

Identification/Verification: Based on the comparison results, the system can perform two main tasks:

- **Face identification**: It attempts to determine the identity of an individual by matching the detected face against a large database of faces. This is useful in scenarios where the identity is unknown and needs to be established.

- **Face verification**: It verifies whether a claimed identity matches the detected face. It compares the detected face against a specific template or reference image associated with a known identity [11].

Face recognition technology finds applications in various fields, including:

- **Security and surveillance**: It can be used for access control systems, surveillance cameras, and law enforcement to identify and track individuals.

- **User authentication**: It can serve as a biometric authentication method for unlocking devices, accessing accounts, or conducting secure transactions.

- **Personalization**: It can be utilized in applications such as social media, targeted advertising, or content recommendation to personalize user experiences based on detected identities.

- **Human-computer interaction**: It enables interaction with computers or devices through facial recognition, such as facial expressions for emotion detection or facial gestures for control.

It is important to note that the deployment and usage of face recognition technology raise concerns about privacy, ethics, and potential biases. Therefore, it is crucial to ensure responsible and ethical practices when implementing and regulating face recognition systems [14].

2.2 How is Face Recognition different from face detection?

Face recognition and face detection are two distinct but related processes within the field of computer vision. Face detection refers to the process of locating and identifying the presence of human faces within an image or video frame. It involves detecting facial regions or bounding boxes that contain faces. The primary goal of face detection is to determine the general location and size of faces within an image or video.

On the other hand, face recognition involves identifying or verifying individuals based on their unique facial features. It goes beyond face detection by analyzing and comparing the specific characteristics of a person's face, such as the arrangement of eyes, nose, mouth, and other distinguishing features [17]. Face recognition focuses on establishing the identity of individuals by matching their faces against a known database of faces.

In summary, the main differences between face recognition and face detection are as follows:

Purpose: Face detection aims to identify the presence of faces in an image or video, while face recognition aims to identify or verify the identity of individuals based on their facial features.

Output: Face detection typically provides the location and size of faces through bounding boxes or facial region coordinates. Face recognition provides the identification or verification of individuals, often in the form of a match with a known identity or a similarity score.

Complexity: Face detection is generally a less complex task as it focuses on detecting face regions based on general visual cues. Face recognition involves more intricate analysis and comparison of facial features to establish identity, requiring advanced algorithms and feature extraction techniques.

Application: Face detection is a fundamental step for many applications that involve further processing of faces, such as face recognition, emotion detection, or face tracking. Face recognition has specific applications where identity verification or identification is required, such as access control systems, law enforcement, or personalization [19].

While face detection is an essential component of face recognition, it is important to understand that face recognition encompasses additional steps beyond detection, involving feature extraction, matching, and identification or verification.

2.3 Different types of face recognition algorithm.

There are various types of face recognition algorithms that have been developed and used in the field. Here are some commonly used types:

Eigenfaces: Eigenfaces is one of the earliest and widely used face recognition algorithms. It utilizes Principal Component Analysis (PCA) to reduce the dimensionality of face images and represent them as a set of eigenfaces. The algorithm then compares the eigenface representation of a test image with known faces to determine the closest match [21].

Fisherfaces: Fisherfaces, also known as Linear Discriminant Analysis (LDA), is another popular face recognition algorithm. It aims to maximize the ratio of between-class scatter to within-class scatter in the feature space. By finding a projection that maximizes class separability, Fisherfaces can effectively discriminate between different individuals.

Local Binary Patterns (LBP): LBP is a texture-based face recognition algorithm that captures local variations in texture patterns within facial regions. It computes binary codes based on the comparison of pixel intensities with neighboring pixels and represents faces using histograms of these codes. LBP has the advantage of being computationally efficient and robust to variations in lighting conditions.

Deep Convolutional Neural Networks (CNN): Deep learning-based approaches, particularly CNNs, have achieved remarkable success in face recognition. CNNs are trained on large datasets to learn hierarchical representations of faces. They can automatically learn relevant features and patterns from raw pixel data, enabling high accuracy in face recognition tasks.

Siamese Networks: Siamese networks are a type of deep learning architecture that learns to compute similarity or dissimilarity between pairs of faces. These networks use two parallel branches that share weights, taking pairs of face images as input [25]. By optimizing the network to minimize the distance between matching pairs and maximize the distance between non-matching pairs, it learns a face embedding space suitable for face verification or identification.

Deep Metric Learning: Deep metric learning approaches focus on learning a suitable metric or distance function in a high-dimensional face feature space. They aim to ensure that faces of the same identity are closer to each other while faces of different identities are farther apart. This facilitates effective face matching and recognition.

It's worth mentioning that these are just a few examples, and the field of face recognition continues to evolve with advancements in machine learning and computer vision. Different algorithms may be suitable for specific scenarios based on factors such as dataset size, computational requirements, and accuracy requirements.

2.4 Two Mode of operations of face recognition

Face recognition technology can generally operate in two modes:

Verification/Authentication Mode: In this mode, face recognition is used to verify or authenticate the identity of an individual. It involves comparing a captured face image with a stored template or reference image to determine if they match. This mode is commonly used for access control systems, such as unlocking smartphones or gaining entry to secure areas [26]. The system confirms whether the person presenting the face is the same as the authorized user or a pre-registered individual.

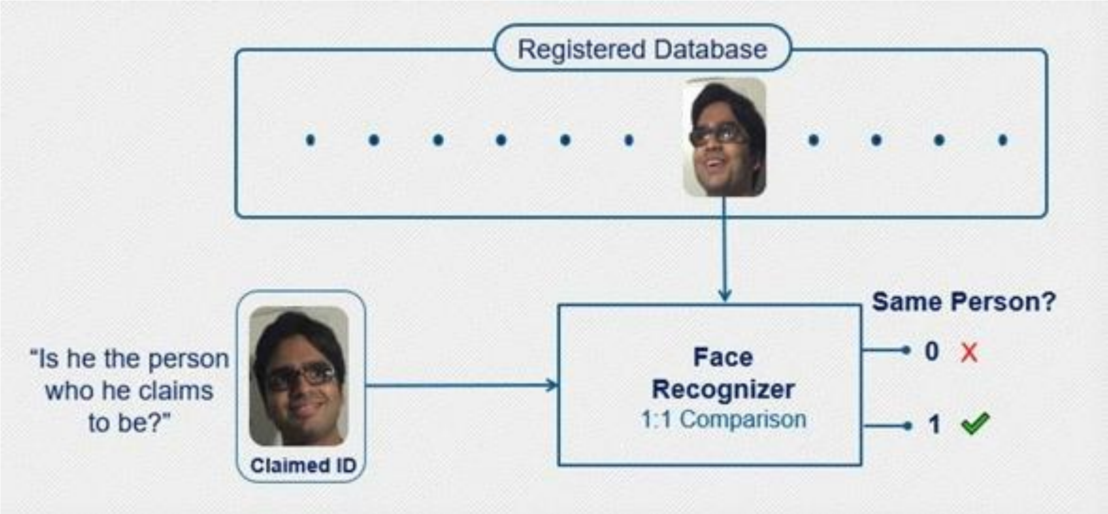

Fig. 2.1: Face Verification

Identification/Recognition Mode: This mode involves the identification of an individual from a large database or a group of people. The face recognition system compares the captured face image with multiple templates or reference images stored in a database to find a potential match. It aims to determine the identity of the person by searching through a collection of known faces [26]. This mode is often used in surveillance applications, such as identifying suspects from CCTV footage or locating missing persons.

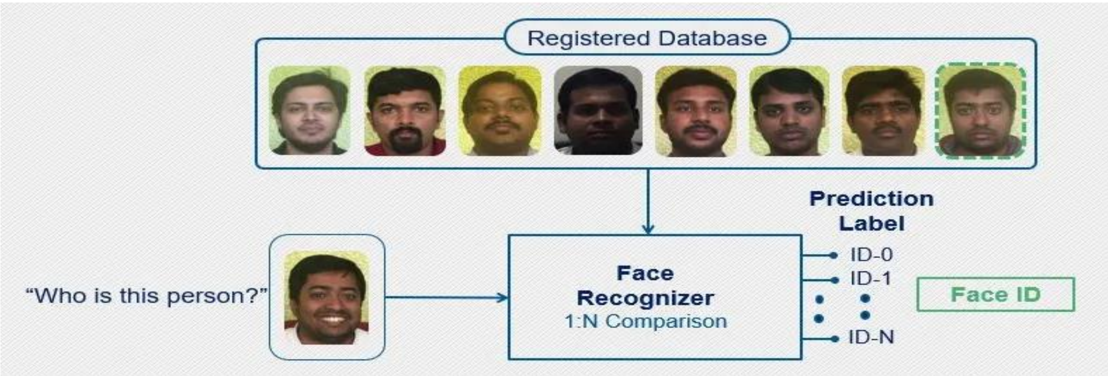

Fig. 2.2: Face Identification

Both verification/authentication and identification/recognition modes rely on similar underlying face recognition algorithms and techniques, but they differ in their objectives and the scale at which they operate [27-28] .

2.5 Different Phases of a face recognition system

A facial recognition system typically involves several stages or steps to process and recognize faces. Here are the various stages commonly found in a facial recognition system:

Face Detection: The first stage is facing detection, where the system locates and identifies faces in an input image or video stream. Face detection algorithms analyze the image to identify regions that potentially contain faces.

Face Alignment: After detecting the face regions, the system may perform face alignment to normalize the face orientations and poses. This step aims to ensure consistent facial representations across different images and reduce the effects of variations in head position and rotation.

Feature Extraction: The next stage is featuring extraction, where distinctive features or descriptors are extracted from the detected and aligned faces. These features capture the unique characteristics of the face and form a compact representation [28]. Popular feature extraction methods for facial recognition include Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), or Deep Convolutional Neural Networks (CNNs).

Feature Encoding: Once the features are extracted, they are typically encoded or transformed into a compact and discriminative representation suitable for recognition. Common encoding techniques include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), or various deep learning-based methods.

Face Matching: In this stage, the system compares the extracted face features or encodings against a database of known faces to find potential matches or similarities. Various similarity metrics or distance measures, such as Euclidean distance or cosine similarity, are used to compute the similarity between face representations [28].

Decision Making: Based on the similarity scores obtained from face matching, a decision is made to determine the identity of the input face. This decision can be a binary classification (e.g., match or non-match) or a multiclass classification (e.g., recognizing specific individuals).

Post-processing and Verification: After the initial decision, post-processing techniques may be applied to refine the recognition results. These techniques can include thresholding, score normalization, or confidence estimation. Additionally, facial recognition systems may incorporate verification steps, such as liveness detection, to ensure that the input is a live face and not a spoof or fake attempt [30].

Database Management: The facial recognition system usually involves managing a database of known faces or enrolled identities. This database stores the face features or encodings of individuals along with their corresponding labels or metadata.

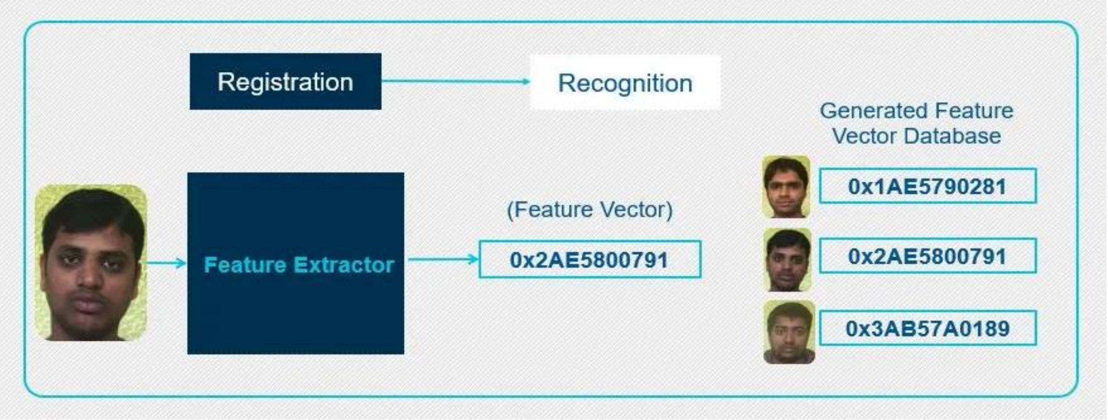

Fig. 2.3: Face Recognition phase I

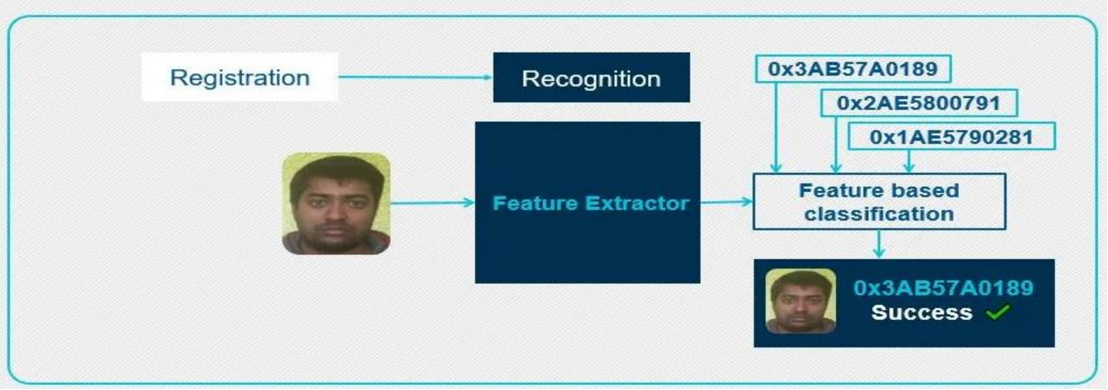

Fig.2.4: Face recognition phase II

An initial operation, also known as registration, involves registering a group of recognized faces. A distinct feature vector is then produced by the feature extractor for each recorded surface. Based on each face's distinctive traits, a feature vector is produced. The extracted feature vector, which is particular to each face, is registered and made available in the database for usage in the future [29].

An input image is delivered to the include extractor within the acknowledgment arrangement to accomplish confront acknowledgment. They also feature an extractor that generates a highlight vector relevant to the supplied confront picture. This include vector is then compared to the feature vectors that are currently accessible in the database. The 'feature-based categorization' component compares the facial attributes of the input face to those of the database's enrolled faces. When an enrolled confront matches the coordinating requirements, the feature-based classification delivers the database's coordinating confront ID [30-31]. It's important to note that different facial recognition systems may have variations in the specific stages or steps involved, depending on the algorithms and techniques used. The stages mentioned above provide a general overview of the common components found in facial recognition systems.

2.6 Components that make up a facial recognition system

The components of a facial recognition system include various components and techniques that work together to enable accurate and reliable face recognition. Here are the essential building blocks:

Data Acquisition: This block involves capturing face images or video frames using cameras or other imaging devices. High-quality and well-lit images are crucial for accurate recognition.

Face Detection: The face detection block locates and identifies faces within the acquired images or video frames. It employs algorithms to identify regions of interest that likely contain faces.

Face Alignment: In this block, the detected faces are aligned to a standardized pose or orientation. Face alignment techniques ensure that facial features are properly aligned, reducing the impact of variations in head pose and rotation [31].

Feature Extraction: This block extracts distinctive features from the aligned faces. These features capture the unique characteristics of the face, such as the shape, texture, and spatial relationships of facial components. Common feature extraction methods include Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), or deep learning-based approaches.

Feature Encoding: Extracted features are encoded or transformed into a compact and discriminative representation. Encoding techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), or deep learning-based methods are used to reduce dimensionality and enhance discriminative power.

Face Database: A face database is maintained to store the feature encodings or representations of known individuals. This database serves as a reference for matching and recognizing faces.

Face Matching: This block compares the feature encodings of the input face with the face database to find potential matches or similarities. Similarity metrics or distance measures, such as Euclidean distance or cosine similarity, are used to compute the similarity between feature representations.

Decision Making: Based on the similarity scores obtained from face matching, a decision is made to determine the identity of the input face. Decision rules or classification algorithms are applied to classify the face as a known identity or an unknown individual.

Post-processing and Verification: Post-processing techniques can be applied to refine the recognition results. These techniques include score normalization, thresholding, or confidence estimation. Additionally, facial recognition systems may incorporate verification steps, such as liveness detection, to ensure the authenticity of the input face.

System Integration: The facial recognition system can be integrated with other applications or systems, such as access control systems, surveillance systems, or identity verification systems, to provide secure and efficient face-based authentication or identification [32].

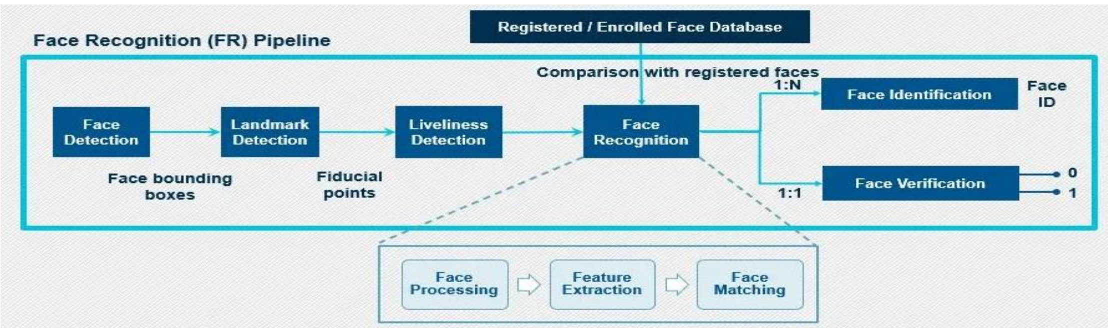

Fig. 2.5: Facial recognition building block

These building blocks collectively form the foundation of a facial recognition system, enabling the capture, detection, alignment, feature extraction, matching, and decision-making processes necessary for accurate and reliable face recognition [33].

CHAPTER 3

Local Binary Pattern (LBP)

Local Binary Pattern (LBP) is a texture descriptor widely used in computer vision and image processing. It was first introduced by Ojala et al. in 1994 as a simple and efficient method for describing local patterns within an image. LBP operates on grayscale images and captures the local structure and texture information [4].

Here's how LBP works:

Image Partitioning: The image is divided into small, overlapping regions called "pixel neighborhoods" or "image patches."

Feature Extraction: For each pixel in the neighborhood, LBP compares the intensity value of the central pixel with its surrounding neighbors. Typically, a circular neighborhood of pixels with a fixed radius is considered.

Binary Representation: LBP assigns a binary value of 1 if the intensity of the neighbor is greater than or equal to the intensity of the central pixel, and 0 otherwise. The binary values of the neighbors are then concatenated to form a binary pattern [6].

Histogram Calculation: After obtaining the binary pattern for each pixel neighborhood, a histogram is created by counting the occurrences of different binary patterns in the image. This histogram represents the distribution of local patterns within the image.

The resulting LBP histogram can be used as a feature vector for various computer vision tasks, such as face recognition, texture classification, and object detection. It encodes information about the texture and structural properties of the image, making it robust to illumination changes and noise [7-9]. LBP has proven to be a computationally efficient and effective texture descriptor, enabling the analysis and recognition of local patterns within an image. It has been widely adopted in many real-world applications due to its simplicity, versatility, and robustness.

3.1 Parameter deployed in LBPH Technique

In Local Binary Patterns Histograms (LBPH), there are a few parameters that can be adjusted to customize the behavior of the algorithm. The main parameters in LBPH are:

Radius used: The radius parameter determines the size of the circular neighborhood around each pixel. It specifies the distance from the central pixel to its neighboring pixels that will be considered for computing the binary pattern. A larger radius includes more pixels in the neighborhood, capturing a wider range of local information, while a smaller radius focuses on a more localized region [10]. The appropriate radius value depends on the scale of the patterns or textures you want to capture.

Number of Points identified: The number of points parameters specifies the number of neighbors to be considered in the circular neighborhood. It defines how many comparisons are made between the central pixel and its surrounding neighbors. Each comparison generates a binary value (0 or 1) based on the intensity relationship between the central pixel and its neighbor. The total number of comparisons determines the length of the resulting binary pattern [12]. A higher number of points captures more detailed local information, but also increases the computational complexity.

Grid Size defined: In some cases, the image is divided into a grid of cells, and LBPH is applied separately to each cell. The grid size parameter determines the number of cells in the x and y directions for the division. Applying LBPH in a grid structure allows capturing spatial variations within the image. The appropriate grid size depends on the size and complexity of the image, as well as the level of detail you want to capture.

These parameters can be adjusted based on the specific application and the characteristics of the images or textures being analyzed. Fine-tuning these parameters can help optimize the performance of LBPH for different tasks, such as face recognition, texture classification, or object detection, by capturing the relevant local patterns and structures [14].

3.2 Training the LBP Algorithm

The Local Binary Patterns (LBP) algorithm itself is not trained in the traditional sense. It is a feature extraction technique that operates on images or image patches. However, to use LBP effectively for a specific task, such as classification or recognition, a separate training process is typically involved. Here's an overview of how you can train a model using LBP features:

Data Collection: Gather a dataset that consists of labeled samples relevant to your task. For example, if you're working on face recognition, collect a dataset of face images with corresponding labels indicating the identity of each person.

Preprocessing: Preprocess the images in your dataset as required. Common preprocessing steps include resizing, normalization, and converting them to grayscale, as LBP operates on grayscale images [16].

LBP Feature Extraction: Apply the LBP algorithm to extract features from the preprocessed images. Compute the LBP histogram or feature vector for each image. The LBP features capture the local texture and structural information of the images.

Feature Representation: Represent each image with its extracted LBP features. This representation could be a histogram of LBP patterns, a concatenated feature vector, or any other suitable representation that summarizes the LBP information.

Train a Classifier: Use the labeled data and their corresponding LBP features to train a machine learning classifier. The choice of classifier depends on your specific task and the nature of the data. Popular choices include Support Vector Machines (SVM), Random Forests, or Neural Networks [16-17]. The classifier learns to map the LBP features to the corresponding class labels.

Model Evaluation: Evaluate the trained classifier on a separate validation or test dataset to assess its performance. Measure metrics such as accuracy, precision, recall, or F1 score to evaluate the effectiveness of the trained model.

Parameter Optimization: Fine-tune the parameters of the LBP algorithm, such as the radius and number of points, and the parameters of the classifier to optimize the performance of the model. This can be done using techniques like cross-validation or grid search.

Deployment: Once you are satisfied with the performance of the trained model, you can deploy it to make predictions on new, unseen data. The LBP algorithm will be applied to extract features from these new images, which are then fed into the trained classifier for prediction or classification [18].

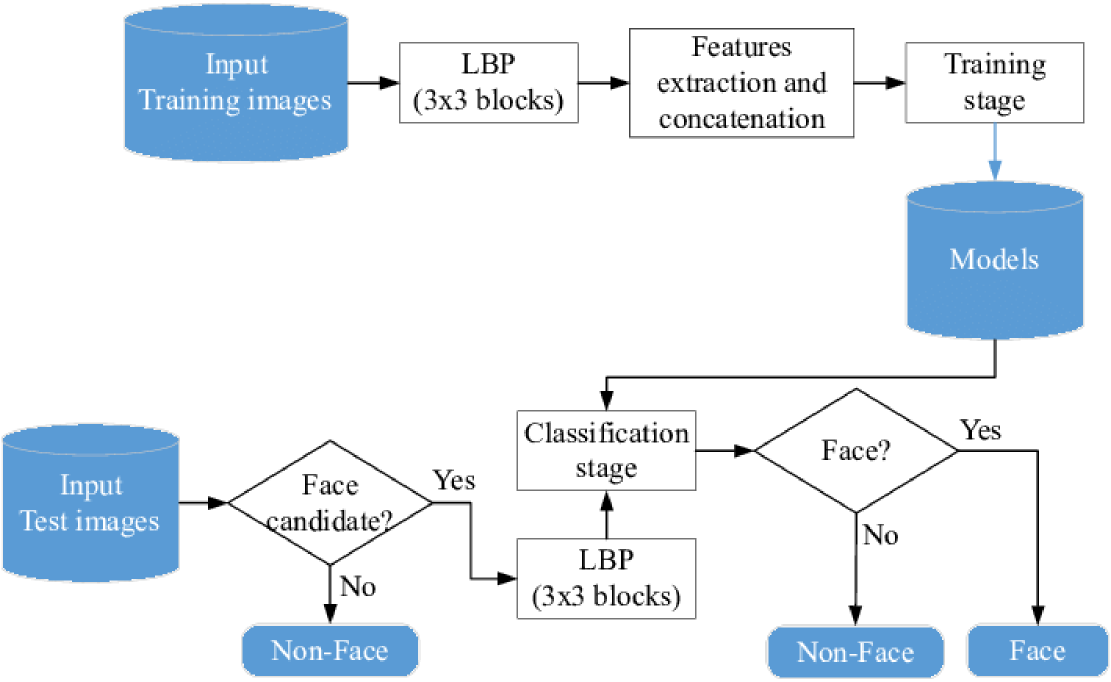

Fig. 3.1: LBP Block Diagram

It's important to note that while the LBP algorithm itself doesn't require training, the overall process described above involves training a separate classifier using the LBP features extracted from your dataset. This enables the model to learn the patterns and relationships between the LBP features and the corresponding labels [20], thereby enabling it to make predictions or classifications on new data.

3.3 LBP Operation

The operation of the Local Binary Patterns (LBP) algorithm involves the following steps:

Image Partitioning: The input image is divided into small, overlapping regions called pixel neighborhoods or image patches [22]. The size of these neighborhoods is determined by the radius parameter.

Center Pixel Selection: For each pixel in a neighborhood, a center pixel is selected. This center pixel acts as the reference point for comparing its intensity with the intensities of its neighboring pixels.

Neighbor Pixel Intensity Comparison: The LBP algorithm compares the intensity value of the center pixel with the intensity values of its neighboring pixels. Typically, a circular neighborhood of pixels is considered, and the neighbors are sampled evenly around the center pixel [23]. The number of neighbors is determined by the number of points parameters.

Binary Pattern Generation: Based on the intensity comparison, a binary pattern is generated for the center pixel. For each neighbor, if the intensity value is greater than or equal to the intensity value of the center pixel, a binary value of 1 is assigned. Otherwise, a binary value of 0 is assigned [24]. The binary values of all the neighbors are concatenated to form the binary pattern.

Histogram Calculation: After generating the binary pattern for each pixel neighborhood, a histogram is computed by counting the occurrences of different binary patterns in the image. The histogram represents the distribution of local patterns within the image.

Feature Extraction: The resulting histogram can be used as a feature vector that captures the local texture and structural information of the image. The histogram bins correspond to different possible binary patterns [26], and the bin values indicate the frequency of occurrence of each pattern.

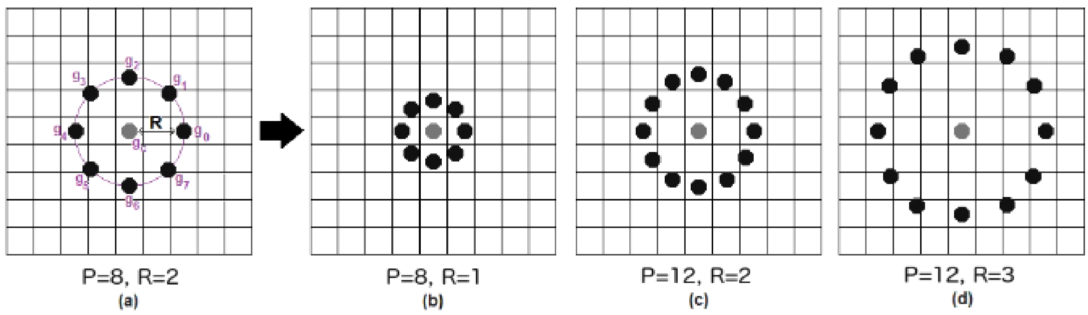

Fig. 3.2: Neighbored set for different P and R

The LBP operation is typically performed on grayscale images, although it can be extended to color images by considering multiple channels separately or by converting the image to a grayscale representation.

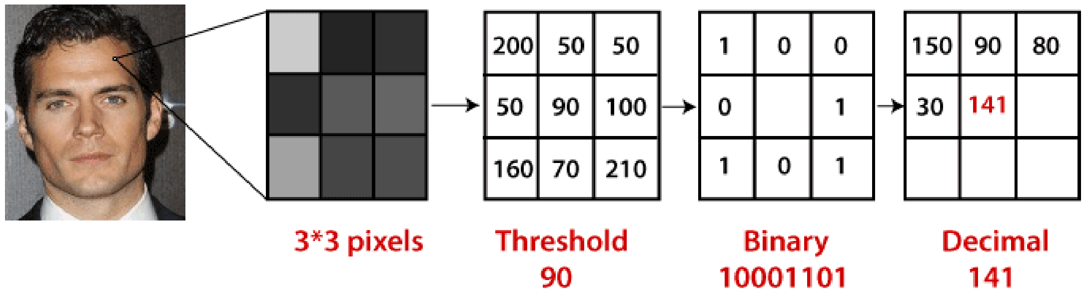

Fig. 3.3: LBP Operation

The LBP algorithm can be applied to the entire image or specific regions of interest, depending on the application. It is known for its simplicity, computational efficiency, and robustness to variations in illumination and noise [26-28]. The resulting LBP features can be used for various tasks, such as texture analysis, face recognition, object detection, and image classification.

3.4 Extracting the Histogram

In LBP-based face recognition, the extraction of the histogram involves the following steps:

Face Detection: The first step is to detect and localize faces in the input image. This can be done using face detection algorithms or pre-trained face detection models.

Face Alignment (Optional): It is often beneficial to align the detected faces to a standardized pose or orientation to reduce the impact of facial variations. This step helps ensure consistent comparisons across different face images [29].

Image Preprocessing: Preprocess the aligned face images, if necessary, to enhance the quality and normalize the appearance. Common preprocessing steps include resizing the images to a fixed size, converting them to grayscale, and applying histogram equalization to improve contrast.

Divide Face into Local Regions: Divide the preprocessed face image into several local regions or sub-regions. These regions can be non-overlapping or overlapping, depending on the desired level of granularity in capturing local facial information [29].

LBP Feature Extraction: Apply the LBP algorithm to each local region of the face image. Calculate the LBP histogram or feature vector for each region. The LBP features capture the local texture patterns and characteristics of the face.

Histogram Concatenation: Concatenate the LBP histograms or feature vectors from all the local regions to form a single combined feature vector for the entire face image. This combined feature vector represents the distribution of local texture patterns across the face [30].

Classification/Matching: Use the extracted feature vector to classify or match the face against a database of known faces. This can be done using various classification algorithms or similarity/distance measures, such as nearest neighbor search or SVM classification.

Model Training and Testing: The LBP-based face recognition system typically involves a training phase and a testing phase. During training, a database of known faces with their corresponding identities is used to train a classifier or build a face recognition model. The LBP features are extracted from the training images, and the model learns to map these features to the corresponding identities [31]. During testing, the trained model is applied to unseen face images to recognize or verify the identities of individuals.

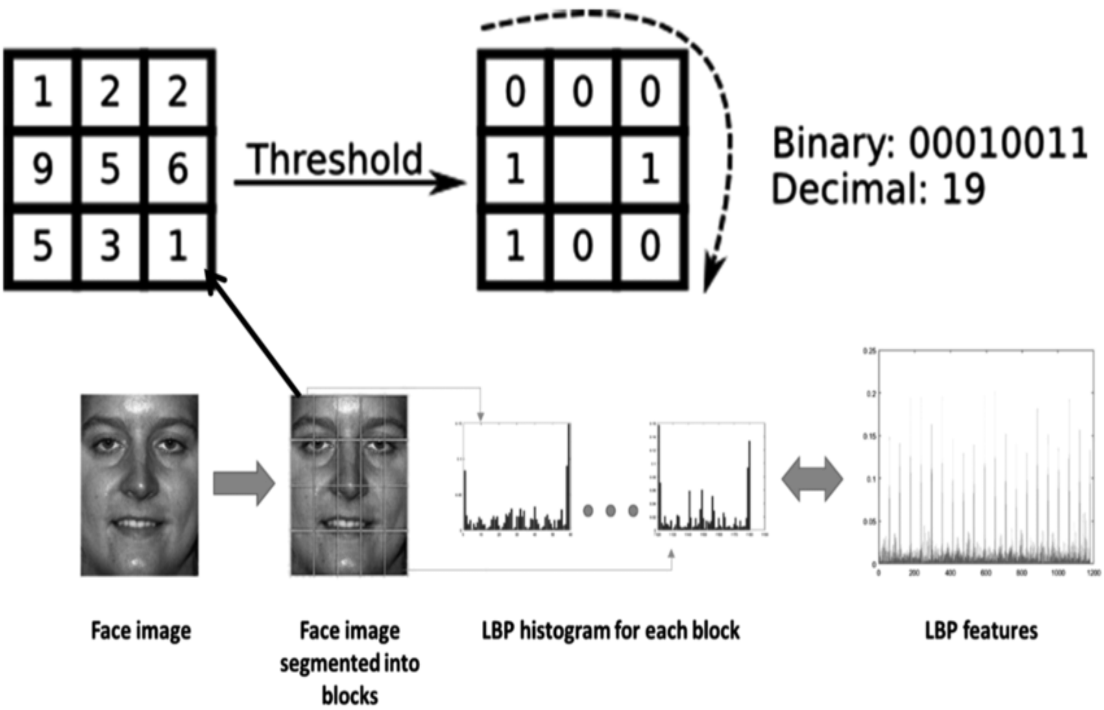

Fig. 3.4: Extracting histogram from an image

The LBP histogram captures the distribution of local texture patterns in different facial regions, providing discriminative information for face recognition. By considering the local texture patterns rather than relying solely on global facial features, LBP-based face recognition can be more robust to variations in lighting, pose, and facial expressions [32].

3.5 performing face Recognition

Performing face recognition involves matching an input face against a database of known faces to determine the identity of the individual. Here's a general outline of the steps involved in performing face recognition:

Face Detection: Detect and locate the face region in the input image or video frame using face detection algorithms. This step identifies the region of interest that potentially contains a face.

Face Alignment (Optional): Align the detected face to a standardized pose or orientation, if necessary, to reduce the impact of variations in head pose and rotation. Face alignment techniques ensure consistent comparisons across different face images.

Feature Extraction: Extract descriptive features from the aligned face region. Common feature extraction methods include Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), or deep learning-based approaches. These features capture the unique characteristics of the face [33].

Feature Encoding: Encode the extracted features into a compact and discriminative representation. Encoding techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), or deep learning-based methods are used to reduce dimensionality and enhance discriminative power.

Face Database: Maintain a database that stores the feature encodings or representations of known individuals [33]. This database serves as a reference for matching and recognizing faces.

Face Matching: Compare the feature encoding of the input face with the face encodings stored in the database. Use similarity metrics or distance measures, such as Euclidean distance or cosine similarity, to compute the similarity between feature representations.

Decision Making: Based on the similarity scores obtained from face matching, make a decision to determine the identity of the input face. This decision can involve thresholding the similarity scores, selecting the closest match, or employing classification algorithms to assign a specific identity label [34].

Post-processing and Verification: Apply post-processing techniques to refine the recognition results. These techniques can include score normalization, thresholding, or confidence estimation. Additionally, incorporate verification steps, such as liveness detection, to ensure the authenticity of the input face.

It's important to note that the performance of face recognition systems heavily depends on the quality of the face detection, the robustness of feature extraction and encoding methods, and the size and diversity of the face database. Continuous evaluation, optimization, and updates to the system are essential for achieving accurate and reliable face recognition results [34].

3.6 Structure of an LBP-based profile face recognition system

An LBP-based profile face recognition system typically consists of the following components and stages:

Data Acquisition: Acquire a dataset of profile face images. These images should capture the side view of individuals' faces.

Data Preprocessing: Preprocess the profile face images to enhance their quality and normalize their appearance. This may involve resizing the images to a fixed size, converting them to grayscale, and applying histogram equalization for better contrast.

Face Localization: Detect and localize the face region within the profile images. This can be done using face detection algorithms or pre-trained face detectors specifically designed for profile faces.

Face Alignment: Align the detected face region to a standardized pose or orientation. This step ensures that the profile faces are properly aligned and reduces the impact of variations in head pose and rotation.

Profile Feature Extraction: Apply the Local Binary Patterns (LBP) operator to the aligned face region to extract the local texture patterns. Calculate the LBP pattern for each pixel by comparing its intensity value with its neighboring pixels. Generate a binary pattern based on these comparisons.

Profile Feature Encoding: Construct histograms of the LBP patterns within the profile face region. Divide the face region into sub-regions or cells and calculate the histogram of LBP patterns within each sub-region. Concatenate or aggregate these histograms to form a combined feature vector representing the profile face [34].

Profile Face Database: Maintain a database that stores the profile face feature vectors. Each feature vector corresponds to a known individual in the database.

Profile Face Matching: Compare the feature vector of the input profile face with the feature vectors stored in the profile face database. Use similarity metrics or distance measures, such as Euclidean distance or cosine similarity, to compute the similarity between feature vectors.

Decision Making: Based on the similarity scores obtained from profile face matching, make a decision to determine the identity of the input profile face. This decision can involve thresholding the similarity scores, selecting the closest match, or employing classification algorithms to assign a specific identity label.

Performance Evaluation: Evaluate the performance of the profile face recognition system using appropriate evaluation metrics such as accuracy, precision, recall, or receiver operating characteristic (ROC) curve. Compare the predicted identities with the ground truth labels to measure the system's effectiveness [34].

Recognition or Verification: Once the profile face recognition model is trained and evaluated, it can be used to recognize or verify the identities of individuals in new unseen profile face images. Extract the LBP features from the input profile face using the same procedure as before. Compare the extracted features with the stored feature vectors of known individuals using distance measures. Make a decision based on the similarity score or apply a threshold to classify the face as a known or unknown identity.

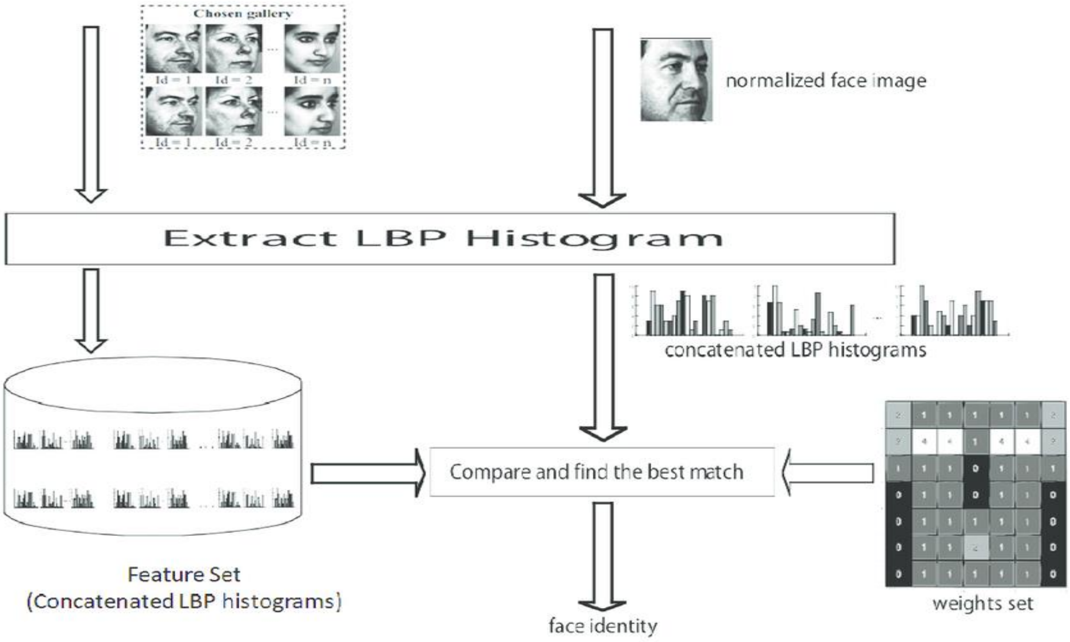

Fig. 3.5: Face identifies from a dataset

It's important to note that the success of an LBP-based profile face recognition system relies on the quality of the face detection and alignment, the selection of LBP parameters (radius, number of neighbors), the design of the feature encoding scheme, and the choice of the classification or matching algorithm. Fine-tuning and optimization may be necessary to achieve accurate and reliable profile face recognition results.

3.7 Advantage of LBP Algorithm

The Local Binary Patterns (LBP) algorithm is a popular feature extraction technique used in computer vision and image processing. It has several advantages, including:

Robustness to Illumination Changes: LBP is highly robust to changes in illumination, making it suitable for applications where lighting conditions can vary. It encodes local texture patterns based on pixel comparisons, rather than absolute intensity values, allowing it to capture meaningful texture information even in the presence of illumination variations [35].

Computational Efficiency: LBP is computationally efficient compared to some other feature extraction techniques. It operates directly on the image pixels and requires minimal preprocessing. The LBP operator can be efficiently implemented using simple bitwise operations, resulting in fast processing times.

Texture Description: LBP is particularly effective in capturing texture information from images. It encodes the local relationships between pixels, which are indicative of different texture patterns. By computing LBP histograms or concatenating LBP patterns from multiple image regions, meaningful texture descriptors can be generated, enabling tasks such as texture classification, segmentation, and retrieval.

Rotation and Scale Invariance: LBP can be made invariant to certain geometric transformations, such as rotation and scale changes. By considering the spatial relationships between neighboring pixels, LBP patterns can be constructed to be insensitive to these transformations, allowing for improved robustness in various applications.

Compact Representation: LBP generates compact feature representations compared to some other methods. The resulting LBP histograms or patterns can be represented using a relatively small number of values, making them memory-efficient and suitable for storage and further processing.

Easy Integration: LBP can be easily integrated with other feature extraction techniques or classification algorithms. It can serve as a complementary feature to color, shape, or other texture descriptors, enhancing the overall representation and improving the performance of various computer vision tasks [35].

Overall, the LBP algorithm provides a powerful and versatile tool for texture analysis and other computer vision applications, offering robustness, efficiency, and compatibility with different image processing techniques.

3.8 Training and Testing

Crop the image to 200*200-pixel values, and reduces the RGB to grayscale with 8 bit per pixels.

100 images generated by algorithm using haarcascade, so 100 histogram is generated.

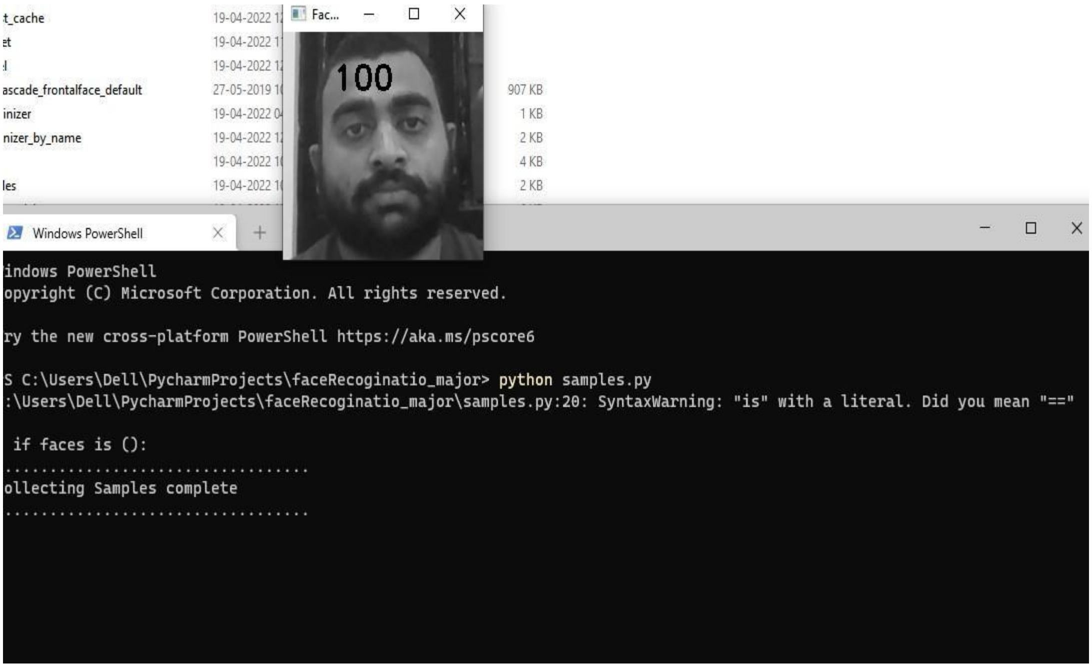

Fig. 3.6: collecting sample for training model

During testing, the algorithm again creates a histogram from the test image, and it compares that with the training histograms to try to get a match.

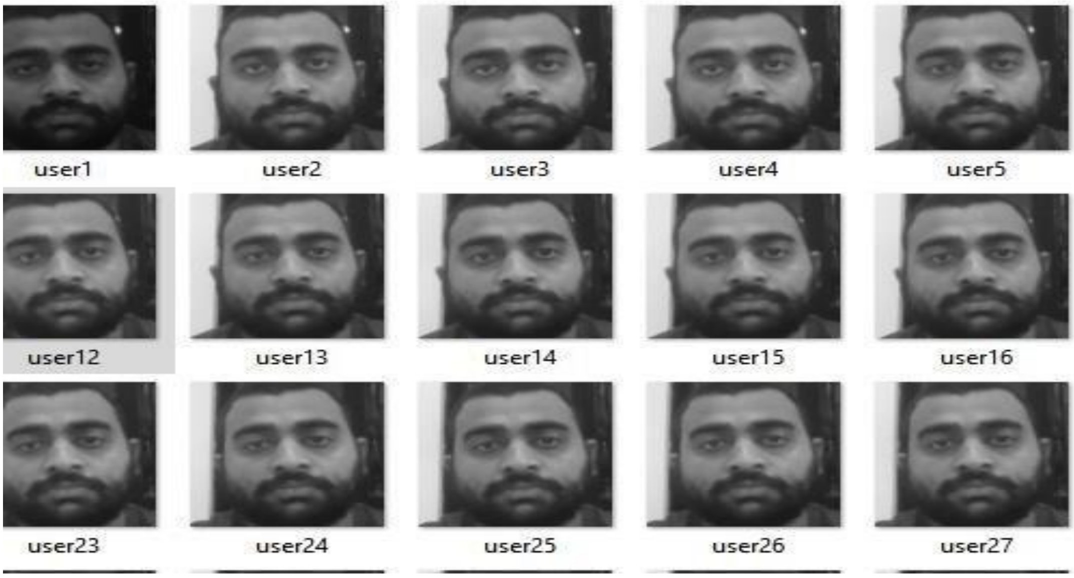

Fig. 3.7: 100 sample data stored

3.9 Result and Evaluation

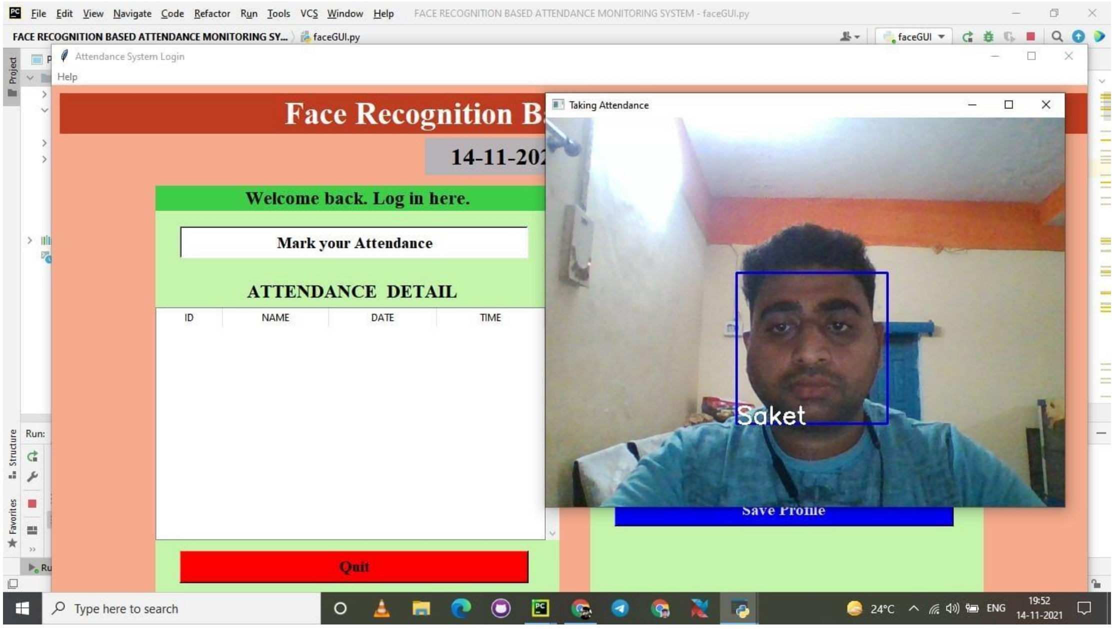

Fig. 3.8: Result

Accuracy	Percentage
Highest	92.5%
Average	90.5%

Table 3.1: Accuracy value of face Recognition in percentage

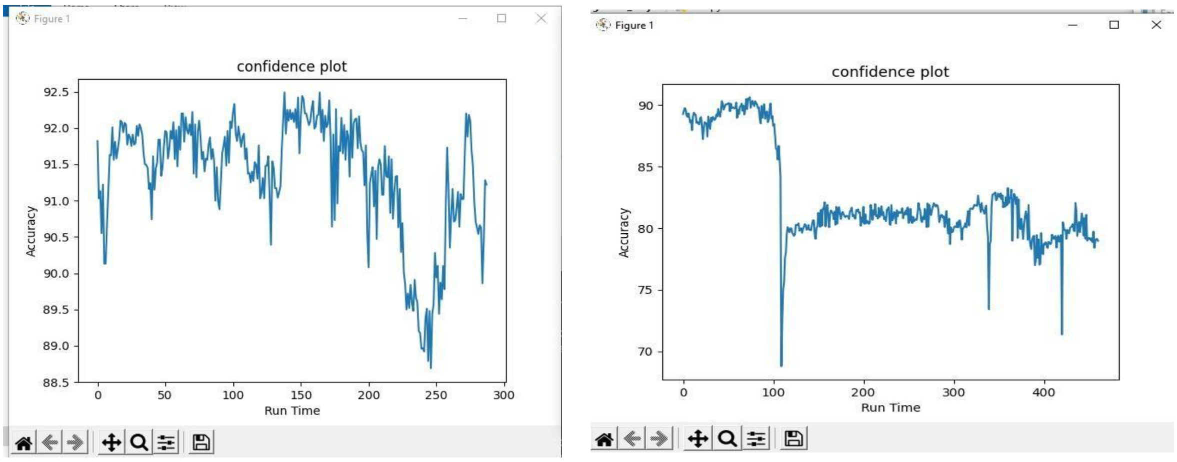

Fig. 3.9: Confidence Plot Graph

CHAPTER 4

Conclusion & Future scope

4.1 Conclusion

In conclusion, the Local Binary Patterns Histogram (LBPH) algorithm is a robust and efficient feature extraction technique for texture analysis and computer vision tasks. It offers several advantages, including its ability to handle illumination changes, computational efficiency, effectiveness in capturing texture information, rotation and scale invariance, compact representation, and easy integration with other techniques.

LBPH has proven to be a valuable tool in various applications, such as texture classification, segmentation, object recognition, and face recognition. Its ability to encode local texture patterns based on pixel comparisons makes it highly suitable for analyzing images with varying lighting conditions. Furthermore, its computational efficiency allows for fast processing, making it applicable to real-time systems.

The rotation and scale invariance properties of LBPH enhance its robustness to geometric transformations, enabling it to handle images taken from different viewpoints or at different scales. This makes it a valuable tool in tasks where object recognition or localization is required. The compact representation of LBPH, achieved through the construction of histograms or concatenation of patterns, allows for memory-efficient storage and processing. Additionally, its easy integration with other feature extraction techniques and classification algorithms makes it a versatile tool that can be combined with other methods to improve overall performance.

The LBPH algorithm is a powerful and flexible approach for extracting texture features from images, offering a range of benefits that make it suitable for a wide array of computer vision applications.

4.2 Future work

As of my last knowledge update in September 2021, the Local Binary Patterns Histogram (LBPH) algorithm is a popular method for facial recognition and texture analysis. However, since I cannot provide information on developments that have occurred after my knowledge cutoff, I can offer some potential directions for future work in the LBPH algorithm based on existing research trends. Please note that these suggestions may or may not have been explored or implemented in the field.

Enhanced Feature Extraction: Investigate alternative ways to extract more discriminative features from local binary patterns (LBP). This could involve exploring different LBP variants, such as uniform patterns, rotation-invariant patterns, or multi-scale patterns. Additionally, combining LBP with other feature extraction techniques, such as deep learning-based methods, may offer improved performance.

Adaptive Neighborhood Selection: Explore methods to adaptively select the neighborhood size for LBP calculation based on the local image characteristics. Different regions of an image may require different neighborhood sizes for effective feature extraction. Adapting the neighborhood selection process could enhance the algorithm's performance in handling variations in image scale and noise.

Robustness to Image Variations: Investigate techniques to make the LBPH algorithm more robust to common image variations such as illumination changes, pose variations, and occlusions. This could involve pre-processing steps to normalize the images, applying advanced normalization techniques, or incorporating robust statistical models to handle variations in the data.

Dimensionality Reduction and Selection: Develop methods to reduce the dimensionality of the feature space while preserving discriminative information. Techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), or other feature selection methods could be explored to improve efficiency and performance.

Fusion with Deep Learning: Explore the integration of LBPH with deep learning models. Deep learning approaches, such as convolutional neural networks (CNNs), have demonstrated remarkable performance in various computer vision tasks. Combining the feature extraction capabilities of LBPH with the representation learning abilities of deep learning could lead to improved recognition accuracy.

Real-Time Implementation: Focus on optimizing the algorithm for real-time applications, particularly on resource-constrained platforms like embedded systems or mobile devices. Investigate techniques to reduce computational complexity, memory requirements, and power consumption without compromising recognition performance.

Privacy and Security: Address privacy and security concerns associated with facial recognition algorithms. Explore methods to enhance the privacy protection of LBPH-based systems, such as incorporating privacy-preserving techniques like secure multiparty computation or differential privacy.

These are just a few potential directions for future work in the LBPH algorithm. Researchers continue to explore and innovate in the field of computer vision, so it is possible that advancements beyond these suggestions have already been made since my knowledge cutoff.

REFERENCE

[1] Abuzneid, Mohannad A., and Ausif Mahmood. "Enhanced human face recognition using LBPH descriptor, multi-KNN, and back-propagation neural network." *IEEE access* 6 (2018): 20641-20651.

[2] Deeba, Farah, Hira Memon, Fayaz Ali Dharejo, Aftab Ahmed, and Abddul Ghaffar. "LBPH-based enhanced real-time face recognition." *International Journal of Advanced Computer Science and Applications* 10, no. 5 (2019).

[3] Ahmed, Aftab, Jiandong Guo, Fayaz Ali, Farha Deeba, and Awais Ahmed. "LBPH based improved face recognition at low resolution." In *2018 international conference on Artificial Intelligence and big data (ICAIBD)*, pp. 144-147. IEEE, 2018.

[4] Stekas, Nikolaos, and Dirk Van Den Heuvel. "Face recognition using local binary patterns histograms (LBPH) on an FPGA-based system on chip (SoC)." In *2016 IEEE International Parallel and* Distributed *Processing Symposium Workshops (IPDPSW)*, pp. 300-304. IEEE, 2016.

[5] Suma, S. L., and Sarika Raga. "Real time face recognition of human faces by using LBPH and Viola Jones algorithm." *International Journal of Scientific Research in Computer Science and Engineering* 6, no. 5 (2018): 6-10.

[6] Kas, Mohamed, Youssef El-merabet, Yassine Ruichek, and Rochdi Messoussi. "A comprehensive comparative study of handcrafted methods for face recognition LBP-like and non LBP operators." *Multimedia Tools and Applications* 79 (2020): 375-413.

[7] Jin, Hongliang, Qingshan Liu, Hanqing Lu, and Xiaofeng Tong. "Face detection using improved LBP under Bayesian framework." In *Third International Conference on Image and Graphics (ICIG'04)*, pp. 306-309. IEEE, 2004.

[8] Mady, Huda, and Shadi MS Hilles. "Face recognition and detection using Random forest and combination of LBP and HOG features." In *2018 international conference on smart computing and electronic enterprise (ICSCEE)*, pp. 1-7. IEEE, 2018.

[9] Rahim, Md Abdur, Md Shafiul Azam, Nazmul Hossain, and Md Rashedul Islam. "Face recognition using local binary patterns (LBP)." *Global Journal of Computer Science and Technology* (2013).

[10] Vázquez, Heydi Méndez, Edel García Reyes, and Yadira Condes Molleda. "A new image division for LBP method to improve face recognition under varying lighting conditions." In *2008 19th International Conference on Pattern Recognition*, pp. 1-4. IEEE, 2008.

[11] Elias, Shamsul J., Shahirah Mohamed Hatim, Nur Anisah Hassan, Lily Marlia Abd Latif, R. Badlishah Ahmad, Mohamad Yusof Darus, and Ahmad Zambri Shahuddin. "Face recognition attendance system using Local Binary Pattern (LBP)." *Bulletin of Electrical Engineering and Informatics* 8, no. 1 (2019): 239-245.

[12] Bah, Serign Modou, and Fang Ming. "An improved face recognition algorithm and its application in attendance management system." *Array* 5 (2020): 100014.

[13] Chinimilli, Bharath Tej, T. Anjali, Akhil Kotturi, Vihas Reddy Kaipu, and Jathin Varma Mandapati. "Face recognition based attendance system using haar cascade and local binary pattern histogram algorithm." In *2020 4th international conference on trends in electronics and informatics (ICOEI)(48184)*, pp. 701-704. IEEE, 2020.

[14] Bai, Xiaojun, Feihu Jiang, Tianyi Shi, and Yuang Wu. "Design of attendance system based on face recognition and android platform." In *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, pp. 117-121. IEEE, 2020.

[15] Preethi, Kolipaka, and Swathy Vodithala. "Automated smart attendance system using face recognition." In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1552-1555. IEEE, 2021.

[16] Chintalapati, Shireesha, and M. V. Raghunadh. "Automated attendance management system based on face recognition algorithms." In *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1-5. IEEE, 2013.

[17] Ahmedi, Aziza, and Suvarna Nandyal. "An automatic attendance system using image processing." *The International Journal Of Engineering And Science (IJES)* 4, no. 11 (2015): 1-8.

[18] Shetty, Anirudha B., and Jeevan Rebeiro. "Facial recognition using Haar cascade and LBP classifiers." *Global Transitions Proceedings* 2, no. 2 (2021): 330-335.

[19] Bhatia, Prayag, Shakti Rajput, Shashank Pathak, and Shivam Prasad. "IOT based facial recognition systemk for home security using LBPH algorithm." In *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*, pp. 191-193. IEEE, 2018.

[20] Srivastav, Gaurav, and Richa Singh. "Facial recognition based workplace security system using LBPH algorithm." In *AIP Conference Proceedings*, vol. 2555, no. 1, p. 040008. AIP Publishing LLC, 2022.

[21] Sayem, Ibrahim Mohammad, and Mohammad Sanaullah Chowdhury. "Integrating face recognition security system with the internet of things." In *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pp. 14-18. IEEE, 2018.

[22] KURNIAWAN, FERRY. "FACE RECOGNITION USING LBPH ALGORITHM WITH MULTIPLE SCENARIO." PhD diss., Universitas Katholik Soegijapranata Semarang, 2023.

[23] Budiman, Andre, Ricky Aryatama Yaputera, Said Achmad, and Aditya Kurniawan. "Student attendance with face recognition (LBPH or CNN): Systematic literature review." *Procedia Computer Science* 216 (2023): 31-38.

[24] Kumar, Ayush, and Deepanshi Singh. "Comprehensive approach of real time web-based face recognition system using Haar Cascade and LBPH algorithm." In *2023 International Conference on Device Intelligence, Computing and Communication Technologies,(DICCT)*, pp. 371-376. IEEE, 2023.

[25] Wangean, Daniel Anando, Sinjiru Setyawan, Fairuz Iqbal Maulana, Gusti Pangestu, and Choirul Huda. "Development of Real-Time Face Recognition for Smart Door Lock Security System using Haar Cascade and OpenCV LBPH Face Recognizer." In *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, pp. 506-510. IEEE, 2023.

[26] Hussain, Sibt Ul, Thibault Napoléon, and Fréderic Jurie. "Face recognition using local quantized patterns." In *British machive vision conference*, pp. 11-pages. 2012.

[27] Biswas, Suparna. "Performance Improvement of Face Recognition Method and Application for the COVID-19 Pandemic." *Acta Polytechnica Hungarica* 19, no. 7 (2022).

[28] Niaraki, Rana Jelokhani, and Asadollah Shahbahrami. "Accuracy improvement of face recognition system based on co-occurrence matrix of local median binary pattern." In *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 141-144. IEEE, 2019.

[29] Jin, Hongliang, Qingshan Liu, Hanqing Lu, and Xiaofeng Tong. "Face detection using improved LBP under Bayesian framework." In *Third International Conference on Image and Graphics (ICIG'04)*, pp. 306-309. IEEE, 2004.

[30] Tan, Xiaoyang, and Bill Triggs. "Fusing Gabor and LBP feature sets for kernel-based face recognition." In *Analysis and Modeling of Faces and Gestures: Third International Workshop, AMFG 2007 Rio de Janeiro, Brazil, October 20, 2007 Proceedings 3*, pp. 235-249. Springer Berlin Heidelberg, 2007.

[31] Qasim, Rameez, M. Mutsaied Shirazi, Naveel Arshad, Ikram Qureshi, and Sajjad Zaidi. "Comparison and improvement of pca and lbp efficiency for face recognition." In *2013 3rd IEEE International Conference on Computer, Control and Communication (IC4)*, pp. 1-6. IEEE, 2013.

[32] Min, Rui, and Jean-Luc Dugelay. "Improved combination of LBP and sparse representation based classification (SRC) for face recognition." In *2011 IEEE International Conference on Multimedia and Expo*, pp. 1-6. IEEE, 2011.

[33] Kas, Mohamed, Youssef El-merabet, Yassine Ruichek, and Rochdi Messoussi. "A comprehensive comparative study of handcrafted methods for face recognition LBP-like and non LBP operators." *Multimedia Tools and Applications* 79 (2020): 375-413.

[34] Kong, Seong G., Jingu Heo, Besma R. Abidi, Joonki Paik, and Mongi A. Abidi. "Recent advances in visual and infrared face recognition—a review." *Computer vision and image understanding* 97, no. 1 (2005): 103-135.

[35] Ghiass, Reza Shoja, Ognjen Arandjelović, Abdelhakim Bendada, and Xavier Maldague. "Infrared face recognition: A comprehensive review of methodologies and databases." *Pattern Recognition* 47, no. 9 (2014): 2807-2824.

Saket Kumar Agarwal(2K21/ISY/21)

by sam user

Submission date: 30-May-2023 02:36AM (UTC-0400)
Submission ID: 2103059118
File name: project.docx (2.53M)
Word count: 6530
Character count: 37919

Saket Kumar Agarwal(2K21/ISY/21)

ORIGINALITY REPORT

9%	5%	4%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.irjmets.com Internet Source	1%
2	Submitted to Eastern Mediterranean University Student Paper	1%
3	dokumen.pub Internet Source	1%
4	Submitted to Akdeniz University Student Paper	1%
5	Submitted to Segi University College Student Paper	1%
6	Submitted to unibuc Student Paper	1%
7	tudr.thapar.edu:8080 Internet Source	<1%
8	link.springer.com Internet Source	<1%
9	dr.ntu.edu.sg Internet Source	<1%