##### Implementation methodology

##### A. Import the required modules

The Modules required to perform the facial recognition are cv2, os, image module and NumPy. cv2 is the OpenCV module and contains the functions for face detection and recognition. OS will be used to maneuver with image and directory names. First, we use this module to extract the image names in the database directory and then from these names individual number is extracted, which is used as a label for the face in that image. Image module from PIL is used to read the image in grayscale format.

##### B. Load the face detection Cascade

To Load the face detection cascade the first step is to detect the face in each image. Once we get the region of interest containing the face in the image, we use it for training the recognizer. For face detection, we will use the Haar Cascade provided by OpenCV. The Haar cascades that come with OpenCV are in the directory of OpenCV installation. haarcascade frontalface default.xml is used for detecting the face. Cascade is loaded using the cv2 CascadeClassifier function which takes the path to the cascade .yml file.

##### C. Create the Face Recognizer Object

The next step involves creating the face recognizer object. The face recognizer object has functions like FaceRecognizer.train() to train the recognizer and FaceRecognizer.predict() to recognize a face. OpenCV currently provides Eigenface Recognizer, Fisherface Recognizer and Local Binary Patterns Histograms (LBPH) Face Recognizer. We have used LBPH recognizer because Real life isn’t perfect. We simply can’t guarantee perfect light settings in your images or 10 different images of a person. LBPH focus on extracting local features from images. The idea is to not look at the whole image as a high-dimensional vector but describe only local features of an object. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighbourhood. LBP operator is robust against monotonic gray scale transformations.

##### D. Prepare the training set and perform the training

To create the function to prepare the training set, we will define a function that takes the absolute path to the image database as input argument and returns tuple of 2 list, one containing the detected faces and the other containing the corresponding label for that face. For example, if the ith index in the list of faces represents the 4th individual in the database, then the corresponding ith location in the list of labels has value equal to 4. Now to perform the training using the Face Recognizer. Train function. It requires 2 arguments, the features which in this case are the images of faces and the corresponding labels assigned to these faces which in this case are the individual number that we extracted from the image names.

**DATASET:**

A collage of a person

Description automatically generated with medium confidence

Fig 3.1 Dataset of a person generated by System

##### E. Testing

For testing the Face Recognizer, we check if the recognition was correct by seeing the predicted label when we bring the trained face in front of camera. The label is extracted using the os module and the string operations from the name of the sample images folder. Lower is the confidence score better is the prediction.

**Used technologies**

#### Face detection

We have used OpenCV which presents a Haar cascade classifier, which is used for face detection. The Haar cascade classifier uses the AdaBoost algorithm to detect multiple facial features. First, it reads the image to be detected and converts it into the gray image, then loads Haar cascade classifier to decide whether it contains a human face. If so, it proceeds to examine the face features and draw a rectangular frame on the detected face. Otherwise, it continues to test the next picture [3]**.**

### Haar features

A simple rectangular Haar-like feature can be defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image. This modified feature set is called 2rectangle feature. Faces are scanned and searched for Haar features of the current stage. The weight and size of each feature and the features themselves are generated using a machine learning algorithm from AdaBoost.

Diagram

Description automatically generated

Fig:Some common Haar Feature (original paper)

#### Face extraction

The LBP operator is applied to describe the contrast information of a pixel to its neighborhood pixels. The original LBP operator is defined in the window of 3\*3. Using the median pixel value as the threshold of the window, it compares with the gray value of the adjacent 8 pixels. If the neighborhood pixel value is larger or equal compared to the median pixel value, the value of pixel position is marked as 1, otherwise marked as 0. The function is defined as shown in equation 1. It can be illustrated in Figure 2.2.

N(x) =Shape

Description automatically generated with medium confidence eq. (1)

Diagram

Description automatically generated with medium confidence

Fig: original LBP operator

**NUMPY AND PANDAS:**

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Open CV**

OpenCV-Python is a library of python bindings designed to solve computer vision problems .open CV makes use of numpy, which is a highly optimized library for numerical operations with a matlab style syntax.all the OpenCV array structures are converted to and numpy arrays. OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.