Automatic Attendance using Face Recognition

(Block Diagram)

Store in Database

Feature Extraction

Scan Faces to train the model

Feature Classification

Update Attendance

Pre Processing

Feature Extraction

Capture Image

Face Detection

**List of Modules**

1. **Face Detection:** Haar Cascade classifier for object detection(here the object is face).
2. **Face Recognition:** LBPH(Local Binary Patterns Histogram) classifier to recognize the faces uniquely.
3. **Feature Extraction:** Principal Component Analysis(PCA) for feature extraction and Euclidian distance classifier.

**Module** : Face Detection

import cv2

import sys

imagePath = "test-data/test.png"

def face\_detect(imagePath):

cascPath = "opencv-files/haarcascade\_frontalface\_alt.xml" #File Downloaded from documentation of opencv

faceCascade = cv2.CascadeClassifier(cascPath)

image = cv2.imread(imagePath)

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

grays = []

faces = faceCascade.detectMultiScale(

gray,

scaleFactor=1.1,

minNeighbors=5,

minSize=(30, 30)

)

for (x, y, w, h) in faces:

cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)

cv2.imshow("Faces found", image)

cv2.waitKey(0)

for i in range(0, len(faces)):

(x, y, w, h) = faces[i]

grays.append(gray[y:y+w, x:x+h])

print("------------------------------------------------------------")

print ("Reading Image -\-")

print("Detecting Face -\-")

print("Face Detection Sucess - Confidence: 100% - Method: HaarCascade")

print(grays)

print("------------------------------------------------------------")

return grays, faces, len(faces)