GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

IMAGE PROCESSING USING DEEP LEARNING

A REPORT FOR EVALUATION 3 OF PROJECT 2

SUBMITTED BY

GYAN PRAKASH

(1613107021/16SCSE107063)

In partial fulfillment for the award degree of

Bachelor of technology

In

Computer science and engineering in Business Analytics

Under the supervision of

Mrs. varsha siasaudia

Professor

April/may 2020

TABLE OF CONTENTS

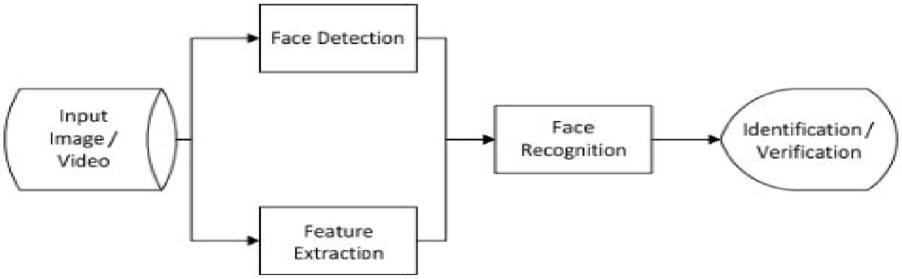
Abstract

Face recognition with different types of data and images is a challenge in the field of computer vision. Some of the major challenges in face recognition are different lying poses, complex background and lightning effects, changing illumination. In the field of computer vision face recognition is one of the relevant and effective applications. In this paper we will discuss different methods, techniques and algorithms which have been used in face recognition and being proposed by many researchers. We will also discuss how, what, and why these methods are used for face recognition and how some of them are very effective. Therefore this paper is all about face recognition and different approaches and methods proposed by many researches.

Keyword: face recognition, biometric application, machine learning, deep learning, image processing.

INTRODUCTION:

Face recognition can be used for real time cloud surveillance, human computer interface, prevention of unauthorised persons from secured devices or any personal file. Several emerging applications, from law enforcement to commercial tasks, demand the industry to develop efficient and automated face recognition systems. Although many researchers have worked on the problem of face recognition for many years still several challenges need to be solved such as different lying poses, illumination effects. Face recognition is one of the biometric methods that has the merits of both high accuracy and low intrusiveness. Biometrics use physiological properties of humans so that without the individual no any other person can steal any information. Recognition of something or someone in the field of computer vision is a computationally challenging task that requires fine discrimination between similarly looking images of different identities, as well as generalization across different images of the same individual. Face recognition has a basic structure which is Image process as input, face detection, preprocessing, face recognition.



Face Detection: The main function of this step is to detect the face from the captured image or the selected image from the database. This face detection process actually verifies that whether the given image has a face image or not, after detecting the face this output will be further given to the pre-processing step.

Pre-processing: This step is working as the pre-processing for face recognition, In this step the unwanted noise, blur, varying lightening condition, shadowing effects can be remove using pre-processing techniques .once we have fine smooth face image then it will be used for the feature extraction process.

Feature Extraction: In this step features of face can be extracted using feature extraction algorithm. Extractions are performed to do information packing, dimension reduction, salience extraction, and noise cleaning. After this step, a face patch is usually transformed into a vector with fixed dimension or a set of fiducial points and their corresponding locations.

Verification: In this step the trained model verifies the given input.

Existing System

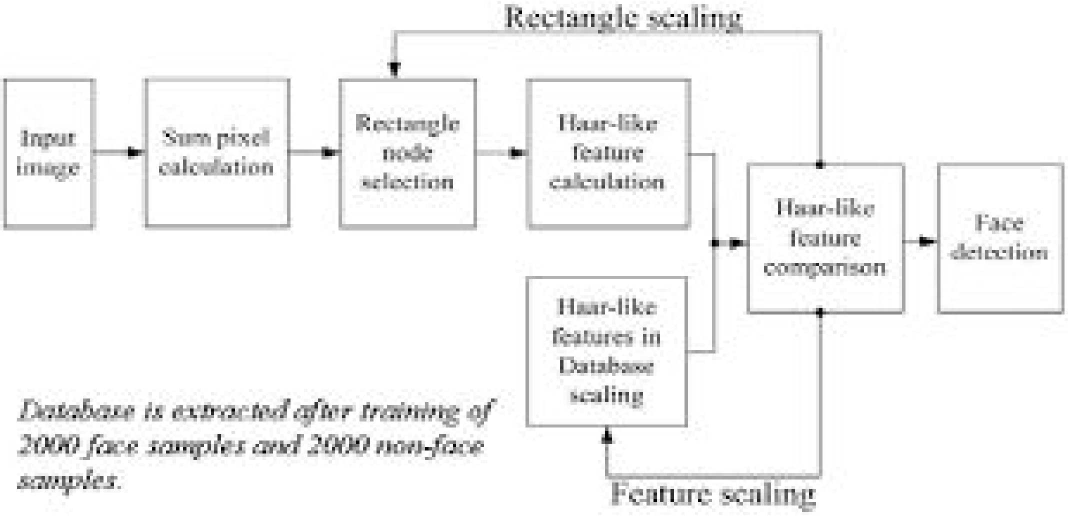
Numeros robust algorithm had been developed and claimed to have accurate performance to tackle face detection and recognition problems . These methods are most successful and widely used for face recognition and detection application.

1. EIGEN FACE :- This method is one of the generally used algorithms for face recognition. The name is given to a set of Eigenvectors in computer vision and developed by Sirovich and Kirby in 1987. It helps in constructing covariance matrices and produces dimension reduction by allowing the smaller set of basis images to represent the original training image. Eigenface can be created by performing Principal component analysis. Principal Component Analysis is used by face recognition and detection. Mathematically, Eigenface are the principal components that divide the face into feature vectors. The feature vector information can be obtained from the covariance matrix. These Eigenvectors are used to quantify the variation between multiple faces. The faces are characterized by the linear combination of highest Eigenvalues.

2. ARTIFICIAL NEURAL NETWORK:- The first method proposed by [navneet jindal] is ANN with PCA(principal component analysis) which is a mathematical model used for classification . In principal component analysis the dimension reduction method and it retain the majority of variation in the dataset. The principal component analysis captures the variation and uses it for face detection . It reduces the high dimensionality image to low dimensionality image to process in the training data for comparing and calculating the variation between all the images which are into the database.

3. CONVOLUTIONAL NEURAL NETWORK :- Convolutional Neural Network Cascade is a method used for Face Detection. It is built with very powerful, and high performance maintainability. The CNN cascade operates at multiple resolutions, and quickly rejects the background regions in the fast low resolution stages, and carefully evaluates a small number of challenging candidates in the last high resolution stage. To improve localization effectiveness, and reduce the number of candidates at later stages, a CNN-based calibration stage is there after each detection stage in the cascade. The motivation of applying the calibration is that the most confident detection window may not be well aligned to the face.

PROPOSED SYSTEM

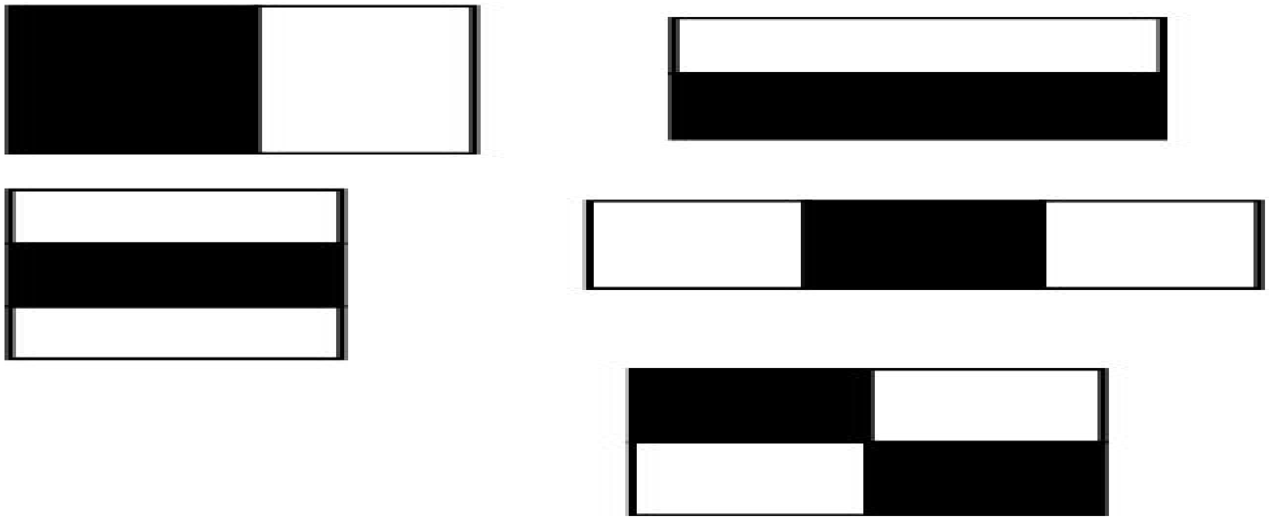

Rectangle scaling
Input image
Sum pixel calculation
Rectangle node selection
Haar-like feature calculation
Haar-like feature comparison
Face detection
Haar-like features in Database scaling
Database is extracted after training of 2000 face samples and 2000 non-face samples.
Feature scaling

The project presents a face recognition system which is used to find the accuracy of the similarity between the given input as well as output. The diagram shows the overall flow process of the face recognition application using haar cascade classification algorithm and PCA.

This system basically works on the basis of some steps : such as image processing , features extraction, output similarity detection.

Haar cascade classifier:- Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.



Haar cascade classifier(classification of eyes , nose and other parts)

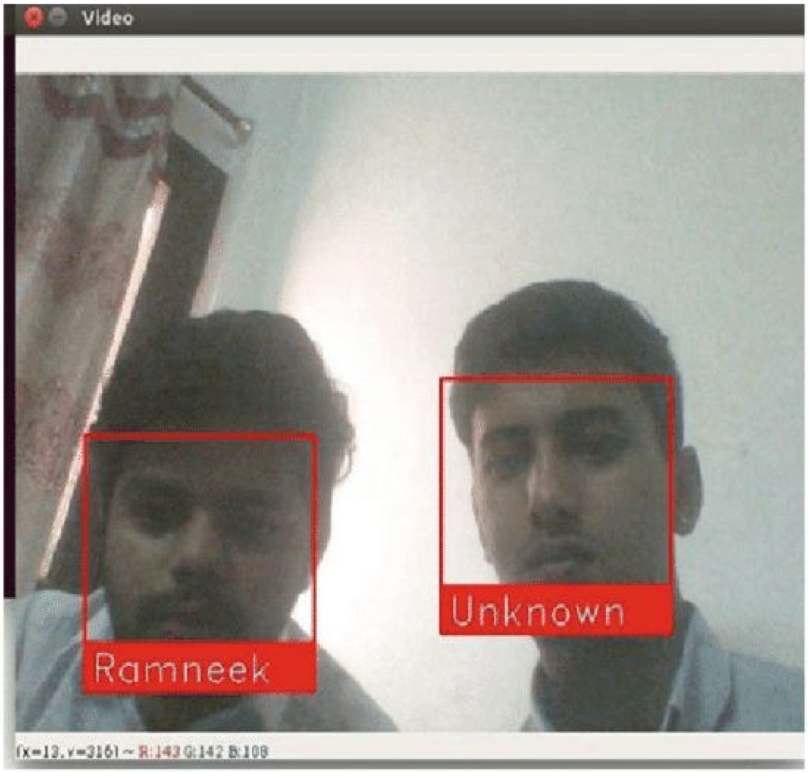
Implementation :-

```
import pickle

import cv2

import numpy as np

import os


recognizer = cv2.face.LBPHFaceRecognizer_create()

recognizer.read('trainer/trainer.yml')

cascadePath = "haarcascade_frontalface_default.xml"

faceCascade = cv2.CascadeClassifier(cascadePath);


font = cv2.FONT_HERSHEY_SIMPLEX
```

```
#iniciate id counter
id = 0

# names related to ids: example ==> Aashish: id=1,  etc

pickle_open=open("name_id_dict.pickle","rb")
name_id_dict=pickle.load(pickle_open)


# Initialize and start realtime video capture
cam = cv2.VideoCapture(0,cv2.CAP_DSHOW)

while True:
    ret, img =cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (30, 30),
       )

    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id1, confidence = recognizer.predict(gray[y:y+h,x:x+w])


        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 60 and id1 in name_id_dict.keys()):
            id = name_id_dict[id1]
            confidence = "  {0}%".format(round(100 - confidence))
        else:
            id = "Unknown"
            confidence = "  {0}%".format(round(100 - confidence))

        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    cv2.imshow('camera',img)

    k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
pickle_open.close()
cam.release()
cv2.destroyAllWindows().
```

The basic implementation is used for the purpose of haar cascade classification of face.

Output



Conclusion:-

This paper includes a summary review of studies which are related to face recognition systems based on different proposed methods and algorithms based on the situation and kind of output we want. In this paper we are discussing different architecture, approach, algorithms, methods, database for training or testing images and performance measures of face recognition system were used in each study. Every researcher has their own approach for recognizing faces from databases or from video. Many researches have tried to solve the problems associated with earlier proposed methods but still there are some advantages and limitations in these discussed methods.