SHRI G. S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE (M. P.)

Topic : Face Recognition System Using Machine Learning Algorithm

Degree : Master's Of Computer Application

Author:

Name : Piyush Sethiya

Enrollment No. – 0801CA221040

Supervisior:

Ass. Prof. Upendra Singh

Face Recognition

Abstract

Face recognition using Machine Learning(sub branch of AI) is a computer vision technology that is used to identify a person or object from an image or video. It uses a combination of techniques including deep learning, computer vision algorithms, and Image processing. These technologies are used to enable a system to detect, recognize, and verify faces in digital images or videos.

Keywords

Face Recognition, Classification Algorithm, Model Based, Python libraries, Machine Learning.

CONTENT:

Chapter 1. Introduction

- ## Basic :

This Project identify a person or object from an image or video. It uses a combination of techniques including deep learning, computer vision algorithms, and Image processing. These technologies are used to enable a system to detect, recognize, and verify faces in digital images or videos.

- ## Aim :

Facial recognition is a way of identifying or confirming an individual's identity using their face. Facial recognition systems can be used to identify people in photos, videos, or in real-time. Facial recognition is a category of biometric security.

- ## Contribution

Dataset :

In this project the Face Recognition dataset. We manually enter the image of a person and that person's face is then detected marked in a rectangular box and then matches its identity to an authorized user.

Parameters :

There are some parameters are use for the Face Recognition system:

- Machine Learning Model Parameter
- Training Dataset
- Perfomance Parameter
- Hyper Parameter

Chapter 2. Literature review

- recognition is an important research problem spanning numerous fields and disciplines. This because face recognition, in additional to having numerous practical applications such as bankcard identification, access control, Mug shots searching, security monitoring, and surveillance system, is a fundamental human behaviour that is essential for effective communications and interactions among people.

Reference : Manuscript received February 22, 2005.

A. S. Tolba is with the Information Systems Department, Mansoura University, Egypt

- Face recognition starts with the detection of face patterns in

sometimes cluttered scenes, proceeds by normalizing the face
images to account for geometrical and illumination changes,
possibly using information about the location and appearance
of facial landmarks, identifies the faces using appropriate
classification algorithms, and post processes the results using
model-based schemes and logistic feedback
.

References: Neupane D, Seok J. A review on deep learning-based approaches for automatic sonar target recognition. Electronics. 2020 Nov 22;9(11):1972.

- A methodology has been described which enables the detection and classification performance of a minehunting sonar to be rigorously measured. Confidence limits can also be placed on the measurements. A method of using a computer model to allow for the effects of the environment is also described. The procedures are suitable for use as contract acceptance trial.

References: A. H. EL-Baz is with the Mathematics Department, Damietta Faculty of

Science, New Damietta, Egypt, and doing PhD research on pattern
recognition

- Developed automated face recognition system to study the potential application for office door access control. The eigenfaces technique based on PCA and artificial neural networks is applied in this system. The training images can be obtained either offline using advance captured and cropped face images, or online using face detection and recognition training modules on the real frontal face images of the system. The system can recognize faces at a reasonable rate and at the distance of 40 cm to 60 cm from the camera with the person's head rotational angle between −20° to +20°. The experimental results also confirm the influences of illumination and pose on the face recognition system.

References : Ibrahim, R. and Zin, Z.M., 2011. Study of automated face recognition system for office door access control application. In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, pp. 132-136. IEEE.

- Performed three experements to improve PCA performance by decreasing the computational time while same performance. A first experiment is performed to find the best number of images for each person to be used in the training set that gives a highest recognition rate. In the second experiment, the analysis is tested using 28 images for each person with 6 images used for training process. the best results if obtained by varying the threshold value. In the third experiment the number of eigenvectors is decreased producing less computation time. The performance in terms of accuracy is same with second experiment with less computational time. This approach reduce the computation time by 35 % compared with the original PCA algorithm particularly with a large database.

References: Abdullah, M., Wazzan, M. and Bo-Saeed, S., 2012. Optimizing face recognition using PCA. arXiv preprint arXiv:1206.1515.

Chapter 3. Methodology

➢ Algorithm -1 : Principal Component Analysis (PCA)

PCA is feature extraction and dimension reduction method, which can be used to solve recognition and compression problems. PCA is a popular linear projection method, and is also known as Eigen space projection, Karhunen and Loeve (KL) transformation, or Hotelling.

Here are brief review how PCA works:

a. Training set of total M images are used to compute the Average Mean as shown in the equation below:

$$AverageMean = \frac{1}{M}\sum_{n=1}^{M} TrainingImages\ (n) \qquad (1)$$

b. Subtract the original image from Average Mean as shown in the equation below:
S=TrainingImages - AverageMean (2)

c. Compute the Covariance Matrix as shown in the equation below:

$$Covariance = \sum_{n=1}^{M} S(n)\, S^T(n) \qquad (3)$$

d. Calculate Eigenvalues and Eigenvectors of the Covariance Matrix.

e. Sort and eliminate Eigenvalues. f. Project the training samples onto the Eigenfaces

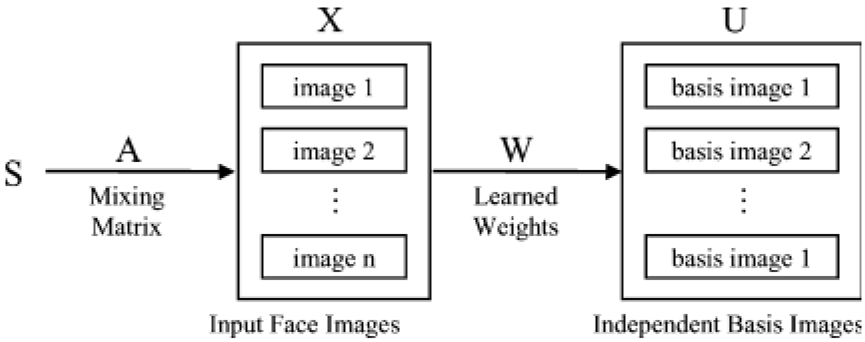
➢ Psuedocode -1

```
1.  # computing the mean
        means=np.mean(training_set,axis=0).reshape(1,10304)
2.  # centering the data
        centered_training_set=training_set-means
3.  # computing the covariance matrix
        covariance_matrix=np.cov(centered_training_set.T,bias=True)
4.  # computing the eigen vectors & eigen values
        eigenvalues,eigenvectors=np.linalg.eigh(covariance_matrix)

5.  # sorting eigen vectors according to their corresponding eigen values
        positions = eigenvalues.argsort()[::-1]

        sorted_eigenvectors = (eigenvectors[:,positions])

        total = sum(eigenvalues)

6.  # getting the required pcs to reach a certain alpha
        r = 0

        current_sum = 0

        while current_sum/total < alpha:
            current_sum += eigenvalues[r]
        r += 1
```

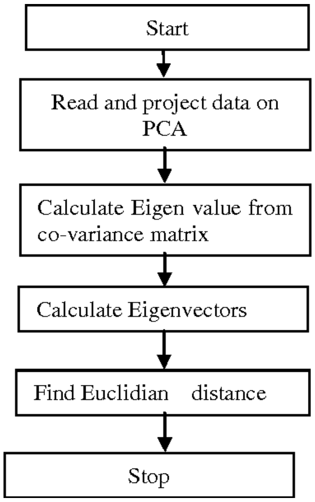
```
7.    # getting the new space that the data will be projected to it
          new_space = eigenvectors[:, :r]

          return new_space
```

- Architecture -1

Figure shows a typical architecture of the PCA.
In this, the output of the Learned Weights is interpreted as the probability of a particular sample belonging to positive class.



- Flowchart -1


Start
Read and project data on PCA
Calculate Eigen value from co-variance matrix
Calculate Eigenvectors
Find Euclidian distance
Stop

Flowchart : PCA

➢ Algorithm -2 – KNN Algorithm:

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data.

Algorithm :

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

- Pseudocode -2

```
1. Load the training and test data
2. Choose the value of K
3. For each point in test data:
    - find the Euclidean distance to all training data points
    - store the Euclidean distances in a list and sort it
    - choose the first k points
    -assign a class to the test point based on the majority of classes present in the chosen points
4. Ends
```

- Flowchart -2

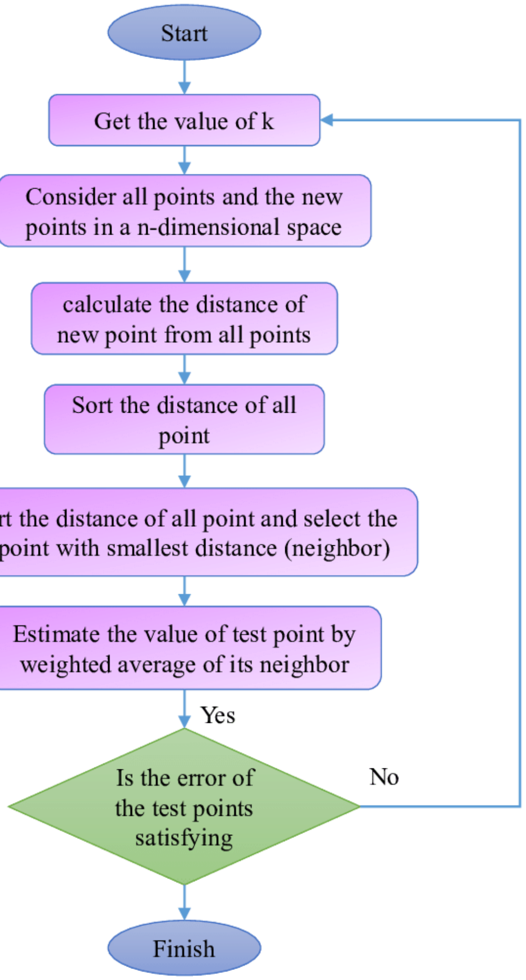

Start
Get the value of k
Consider all points and the new points in a n-dimensional space
calculate the distance of new point from all points
Sort the distance of all point
Sort the distance of all point and select the k point with smallest distance (neighbor)
Estimate the value of test point by weighted average of its neighbor
Yes
Is the error of the test points satisfying
No
Finish

Fig : KNN algorithm flowchart

Chapter 4. Implementation

➢ Hardware:

1) Developer’s Side :

- RAM : 8 GB
- System Type : 64-bit operating system
- Processor : Intel Core i5 , x64-based processor

➢ Software

- Environment : Jupyter Notebook and VS-code
- Operating System : It uses both windows environment
- Language used : Python3

❖ Python Libraries with detailed:

OpenCV library : is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

Face Recognition Library : Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark. This also provides a simple `face_recognition` command line tool that lets you do face recognition on a folder of images from the command line!

- **Dataset description:**

In this project the Face Recognition dataset. We manually enter the image of a person and that person's face is then detected marked in a rectangular box and then matches its identity to an authorized user.

- **Source Code with description**

-
  ```
  # Import the necessary packages
  import cv2 as cv
  import face_recognition
  import matplotlib.pyplot as plt
  ```

-
  ```
  # Load the known image
  known_image = face_recognition.load_image_file("pawankrgunjan.jpeg")
  known_faces = face_recognition.face_encodings(face_image = known_image,
  num_jitters=50,
  model='large')[0]
  ```

-
  ```
  # Lanch the live camera
  cam = cv.VideoCapture(0)
  ```

-
  ```
  #Check camera
  if not cam.isOpened():
  print("Camera not working")
  exit()
  ```

-
  ```
  # when camera is opened
  while True:
  ```

```
➢ # campture the image frame-by-frame
  ret, frame = cam.read()


➢ # check frame is reading or not
  if not ret:
  print("Can't receive the frame")
  break


➢ # Face detection in the frame
  face_locations = face_recognition.face_locations(frame)


  ]for face_location in face_locations:
  top, right, bottom, left = face_location
➢ # Draw a rectangle with blue line borders of thickness of 2 px
  frame = cv.rectangle(frame, (right,top), (left,bottom), color = (0,0, 255), thickness=2)
➢ # Check the each faces location in each frame
  try:
➢ # Frame encoding
  Live_face_encoding = face_recognition.face_encodings(face_image = frame,
  num_jitters=23,
  model='large')[0]


➢ # Match with the known faces
  results = face_recognition.compare_faces([known_faces], Live_face_encoding)


  if results:
  img = cv.cvtColor(frame, cv2.COLOR_BGR2RGB)
  img = cv.putText(img, 'PawanKrgunjan', (30, 55),
  cv2.FONT_HERSHEY_SIMPLEX, 1,
  (255,0,0), 2, cv2.LINE_AA)
  print('Pawan Kumar Gunjan Enter....')
  plt.imshow(img)
  plt.show()
  break
  except:
```

img = cv.putText(frame, 'Not PawanKrgunjan', (30, 55),
cv2.FONT_HERSHEY_SIMPLEX, 1,
(255,0,0), 2, cv2.LINE_AA)

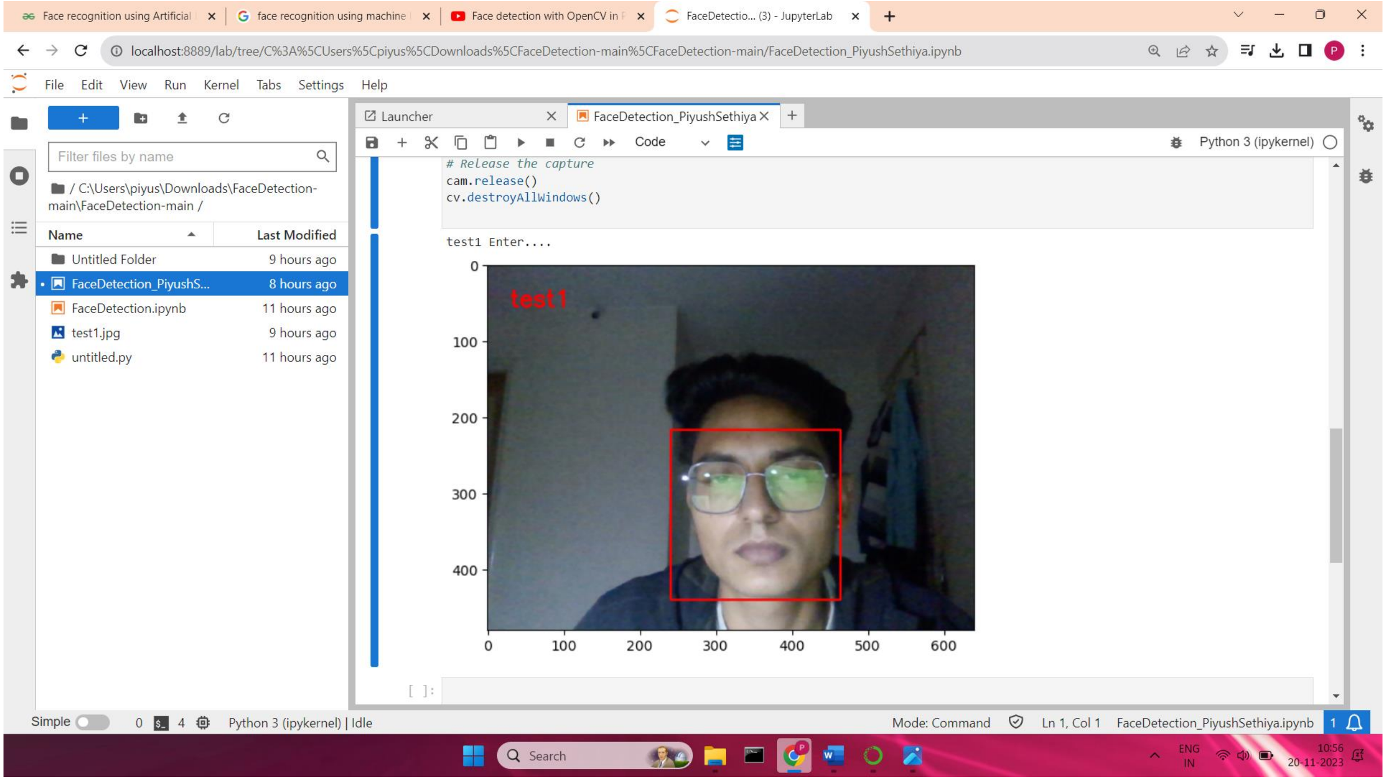
- # Display the resulting frame
  cv.imshow('frame', img)
- # End the streaming
  if cv.waitKey(1) == ord('q'):
  break

- # Release the capture
  cam.release()
  cv.destroyAllWindows()

Chapter 5. Result

❖ Result figure generated by source code with description-

We manually enter the image of a person and that person's face is then detected marked in a rectangular box and then matches its identity to an authorized user.



```
# Release the capture
cam.release()
cv.destroyAllWindows()
```

```
test1 Enter....
```

Name	Last Modified
Untitled Folder	9 hours ago
FaceDetection_PiyushS...	8 hours ago
FaceDetection.ipynb	11 hours ago
test1.jpg	9 hours ago
untitled.py	11 hours ago

Face recognition using Artificial
face recognition using machine
Face detection with OpenCV in
FaceDetectio... (3) - JupyterLab
localhost:8889/lab/tree/C%3A%5CUsers%5Cpiyus%5CDownloads%5CFaceDetection-main%5CFaceDetection-main/FaceDetection_PiyushSethiya.ipynb
File
Edit
View
Run
Kernel
Tabs
Settings
Help
Filter files by name
/ C:\Users\piyus\Downloads\FaceDetection-main\FaceDetection-main /
Launcher
FaceDetection_PiyushSethiya
Code
Python 3 (ipykernel)
test1
0
100
200
300
400
0
100
200
300
400
500
600
[ ]:
Simple
0
4
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
FaceDetection_PiyushSethiya.ipynb
Search
ENG
IN
10:56
20-11-2023

Chapter 6. Conclusion

Our project “Face Recognition System by the evaluation of machine learning algorithms” is used to recognize face of an authorized person. Now the world becomes more and more better because of the advance in science and technology, so face recognition is slowly recognized by people, and we also began to use it in different fields. Face recognition is the use of human facial features to complete identification.

References:

1. **https://www.youtube.com/watch?v=qUI7lOv7-4U**
2. **https://www.geeksforgeeks.org/face-recognition-using-artificial-intelligence/**
3. **https://www.javatpoint.com/face-recognition-in-mchine-learning**