Face Recognition

Algorithms Used:

LBPH :

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

1.Parameters

- Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
- Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
- Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
- Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

Cascade classifier :

Cascading classifiers are trained with *positive* sample views of a particular object and arbitrary *negative* images of the same size. After the classifier is trained it can be applied to a region of an image and detect the object in question. To search for the object in the entire frame, the search window can be moved across the image and check every location for the classifier. This process is most commonly used in image processing for object detection and tracking, primarily facial detection and recognition.

1.Parameters

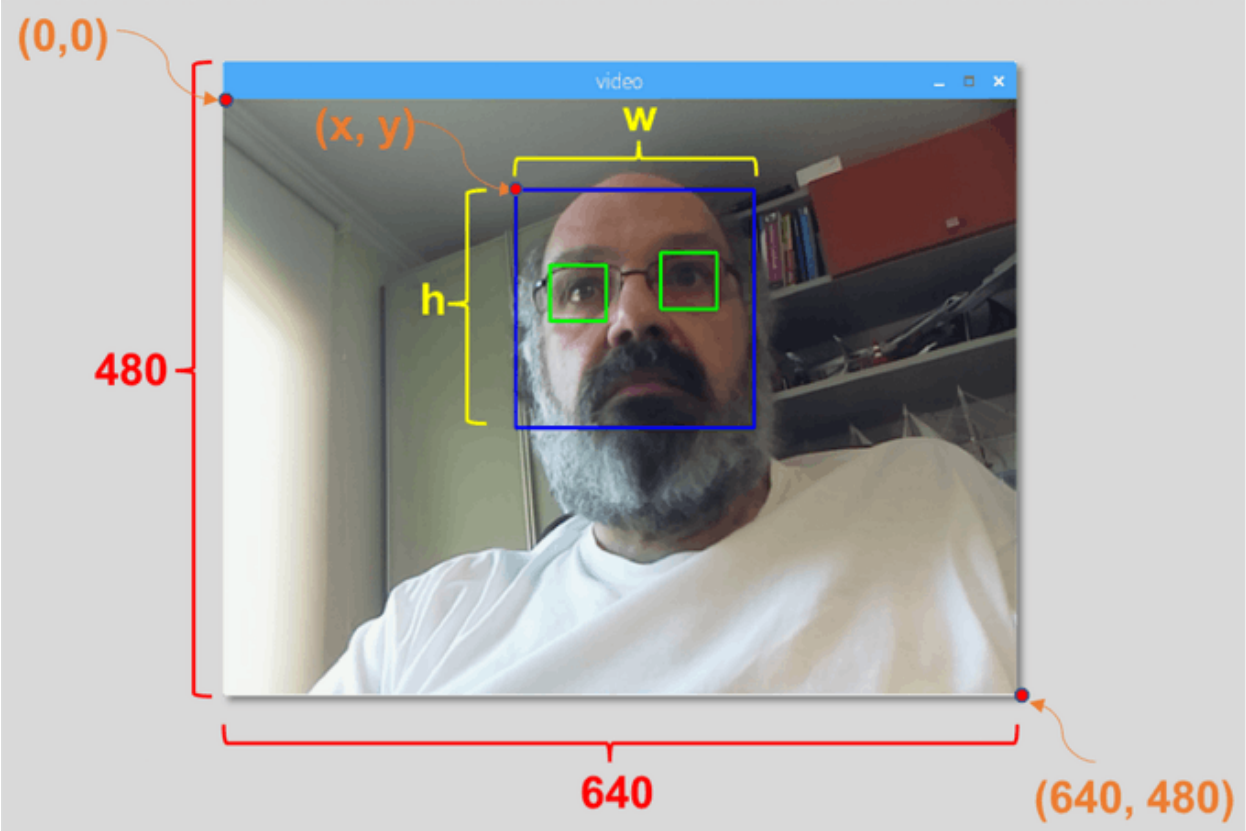
- Scale factor - Parameter specifying how much the image size is reduced at each image scale.
  Basically, the scale factor is used to create your scale pyramid. More explanation, your model has a fixed size defined during training, which is visible in the XML. This means that the size of the face is detected in the image if present. However, by rescaling the input image, you can resize a larger face to a smaller one, making it detectable by the algorithm.
  1.05 is a good possible value for this, which means you use a small step for resizing, i.e. reduce the size by 5%, you increase the chance of a matching size with the model for detection is found. This also means that the algorithm

works slower since it is more thorough. You may increase it to as much as 1.4 for faster detection, with the risk of missing some faces altogether.

- minNeighbors – Parameter specifying how many neighbors each candidate rectangle should have to retain it.

  This parameter will affect the quality of the detected faces. Higher value results in fewer detections but with higher quality. 3~6 is a good value for it.

2. If faces are found, it returns the positions of detected faces as a rectangle with the left up corner (x,y) and having “w” as its Width and “h” as its Height ==> (x,y,w,h). Please see the picture.



Data Augmentation:

Albumentations is a Python library for fast and flexible image augmentations. Albumentations efficiently implements a rich variety of image transform operations that are optimized for performance, and does so while providing a concise, yet powerful

image augmentation interface for different computer vision tasks, including object classification, segmentation, and detection.

File Management:

The attendance csv file is created for each day. We have used exception handling to create the new file.

This repository contains:

1. Existing_Data file :

This python file uses all the previously stored images to generates the primary training data(gray scaling and face detection).
Now, augmentation of each image is done to increase the training data set.
Also, it sortes all the id's into the name.txt file

2. Add_new_employee file:

This python file is used to take a single image of a new data set.
This image is then augmented and stored into the training dataset.
Also, appends the employee id into the name.txt file.

3. Face_traing file:

This python file is used to train the LBPH model and then storing the trained model in trainer.yml file.
This allows us to add only one new set of augmented images rather than generating augmented images of all the existing members.
Trained model is stored to reduce the time taken during prediction.

4. face_recognition file :

This python file is used to identify and recognize the face captures in real time video.
Different csv files are created everyday.
Id, Date and entry time is added to the csv file.

5. Attendance folder :

Contains all the csv files.

6. Dataset folder :

Contains training data formed after augmentation.

7. Img folder:

Raw training data

8. Trainer folder:

trainer.yml: file where trained model is stored

Steps to execute the program :

1. Adding of data
   1.1 Adding a single new output:
   - File - Add_new_employess
   - Run the 1st Cell i.e. used for initializing the parameters for augmentation.
   - We have used an albumentation library to transform the experiment data by flipping,changing contrast,brightness etc.
   - Run the 2nd cell to take the input. Re-run the 2nd cell until the satisfied image is not captured.
   - Run the 3rd cell where augmentation of the image is done and these are added to the dataset.

   1.2 For Existing data
   - File - Existing_Data
   - Run all the cells where
   - 1st cell is used for initializing the parameters for augmentation
   - 2nd cell is used for generating primary grayscale training data.We have used cascade classifier from OpenCV library to detect faces in gray scale.
   - 3rd cell is used for augmentation of all the primary training data and adding it to the dataset.

2. Training of Model
   - File - face_training
   - Run the cell.
   - This cell trains the model and stores the model in the trainer.yml file.
   - Algorithm used for training the model is LBPH(Local Binary Pattern Histogram)

3. Face identification
   - File - face_recognition
   - Run this cell.
   - It will activate the webcam of your device and detect the faces.
   - We have used the VideoCapture() function from OpenCV to read video footage.
   - If the faces are detected then their id along with date and time will be added to csv file.
   - Press Esc to close video capture.