Artificial Intelligence: Theory and Practice

EC3066D

Face Recognition

using HOG and Facenet

Melvin Manoj B200923EC batch: EC03
Salvin Saji B200030EC batch: EC04
Group : 11

Face Recognition

Facial recognition is an area of computer vision and machine learning that has seen significant growth and development in recent years. It involves analyzing and recognizing patterns in an image to identify and verify the identity of an individual. This technology has found widespread applications in security systems, surveillance, and biometric authentication.

Problem Statement:
Automate the attendance-taking procedure in classrooms. Take a photo of the classroom, upload it to the server and the attendance of each student should be automatically marked by the underlying face recognition algorithm.

Objectives:
The goal of this project is to develop a highly accurate and efficient facial recognition system that can be applied in a variety of real-world scenarios. By utilizing the HOG algorithm and neural networks, we aim to overcome the limitations of traditional feature extraction methods and provide a robust solution for facial recognition.

Literary Survey:
There are numerous techniques that can be employed to do face recognition. Some of the methods are listed below with a brief description.

Eigenfaces:
Eigenfaces, also known as Principal Component Analysis (PCA), is a widely used method for face recognition. This method works by projecting face images onto a lower-dimensional space, where the projection coefficients are used as feature vectors. The eigenvectors of the covariance matrix of the face images are computed, and the projections of the face images onto these eigenvectors are used as the feature vectors. The method has been shown to be effective for face recognition in controlled environments, but can struggle with variations in lighting, pose, and facial expressions. However, it is computationally efficient and easy to implement.

Local Binary Patterns (LBP):
Local Binary Patterns (LBP) is a texture-based method for face recognition that operates on grayscale images. This method works by comparing the pixel values in a small neighborhood around each pixel to the value of the center pixel. The resulting binary string is then used as a feature vector. LBP has been shown to be robust to variations in lighting and pose, and is computationally efficient. It is particularly effective when combined with other techniques, such as PCA or SVMs.

Gabor Wavelets:
Gabor wavelets are a set of filters that are designed to capture both spatial and frequency information in an image. Gabor wavelets have been used for face recognition by extracting features from the face image using a bank of Gabor filters. The resulting feature vector is then used for classification. Gabor wavelets have been shown to be effective for face recognition in controlled environments, but can struggle with variations in lighting and pose.

Histogram of Oriented Gradients (HOG):
HOG is a method for feature extraction that captures information about the distribution of edge orientations in an image. HOG has been used for face recognition by first encoding a picture using the HOG algorithm to create a simplified version of the image. Using this simplified image, the part of the image that most looks like a generic HOG encoding of a face is found. The pose of the face is then determined by finding the main landmarks in the face, and the image is warped so that the eyes and mouth are centered. The resulting image is passed through a neural network that knows how to measure features of the face, and 128 measurements are saved. Looking at all the faces measured in the past, the face with the closest measurements is identified as the match.

Deep Learning:
Deep learning methods, particularly Convolutional Neural Networks (CNNs), have recently achieved state-of-the-art performance in face recognition. These methods work by learning hierarchical representations of face images through multiple layers of convolutions and pooling operations. The learned representations are then used as feature vectors for classification. Deep learning methods have been shown to be highly effective for face recognition in both controlled and uncontrolled environments, and can handle variations in lighting, pose, and facial expressions. However, these methods require large amounts of data and computational resources for training, and may not be as easily interpretable as traditional methods like Eigenfaces and LBP.

3D Face Recognition:
3D face recognition methods use 3D models of faces to extract features for recognition. These methods can provide robustness to variations in lighting and pose, and can also capture fine-grained facial details. However, they require specialized hardware for 3D scanning, and are more computationally expensive than 2D methods.

In summary, face recognition is a well-researched area with a variety of methods and techniques. While traditional methods like Eigenfaces and LBP remain relevant, deep learning methods have recently achieved state-of-the-art performance. Gabor wavelets and HOG provide alternative approaches for feature extraction. This project uses HOG.

Background:

The different components in this project can be divided as follows:

Encoding the Image using HOG:
We first convert the input image into a simplified version using the HOG algorithm. This algorithm divides the image into small cells, and computes the gradient orientation of each cell. The orientation values are then used to create a histogram of oriented gradients, which represents the shape and texture of the image. The resulting HOG feature descriptor is used to identify the face in the image.

Finding the Face Landmarks:
After extracting the HOG features from the image, we use the simplified version to identify the part of the image that most closely resembles a generic HOG encoding of a face. Then, we find the main landmarks of the face, such as the eyes, nose, and mouth. We use these landmarks to warp the image so that the eyes and mouth are centered, which helps to improve the accuracy of the recognition algorithm.

Feature Extraction using Neural Networks:
Once the face is centered, we pass the image through a neural network that has been trained to measure features of the face. This neural network extracts 128 feature measurements from the image, which are then saved for comparison with other faces.

Matching with the Database:
We compare the 128 feature measurements extracted from the input image with those of previously saved faces in the database. The face that has the closest measurements to the input image is considered the match.

Methodology:

The HOG algorithm

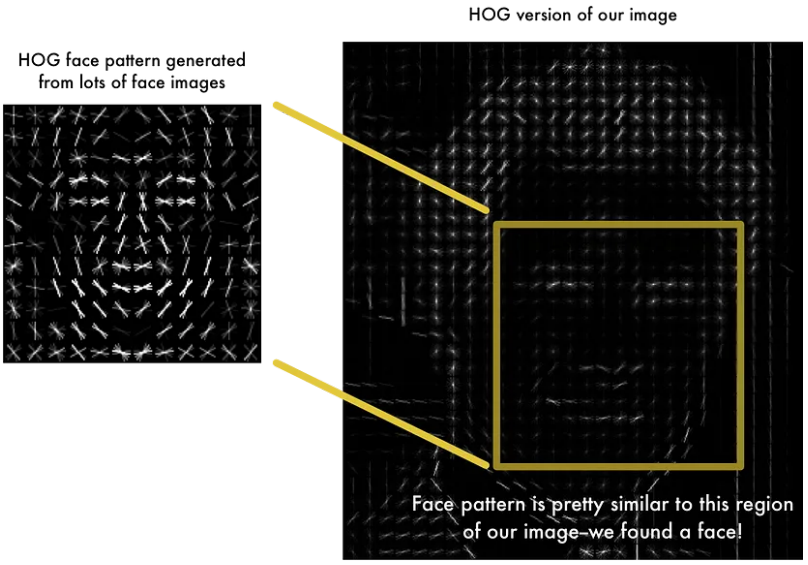
The Histogram of Oriented Gradients (HOG) algorithm is a popular method for feature extraction in computer vision. The HOG algorithm works by dividing the input image into small cells, computing the gradient orientation of each cell, and using these orientations to create a histogram of gradient orientations. The resulting histogram represents the shape and texture of the image.

The gradient orientation of an image is the direction of the steepest change in intensity. To compute the gradient orientation, we use a filter, such as a Sobel filter, to calculate the horizontal and vertical gradients of the image. The magnitude of the gradient is the square root of the sum of the squares of the horizontal and vertical gradients, while the orientation of the gradient is the arctangent of the vertical gradient divided by the horizontal gradient.

To create the histogram of oriented gradients, we first divide the image into small cells of a fixed size. For each cell, we calculate the gradient orientation of the pixels within the cell and create a histogram of the orientations. The orientations are binned into a fixed number of bins, such as 9, and the histogram represents the distribution of the gradient orientations in the cell.

To create a feature descriptor for the image, we then combine the histograms for each cell to form a block. We slide the block across the image, overlapping the blocks by a certain percentage, and concatenate the histograms of each block to form the final feature descriptor.

The HOG algorithm is particularly useful for detecting and recognizing objects in an image because it captures information about the shape and texture of the object. It has been successfully applied to a wide range of tasks, such as object detection, pedestrian detection, and facial recognition.



Finding the Face Landmarks:

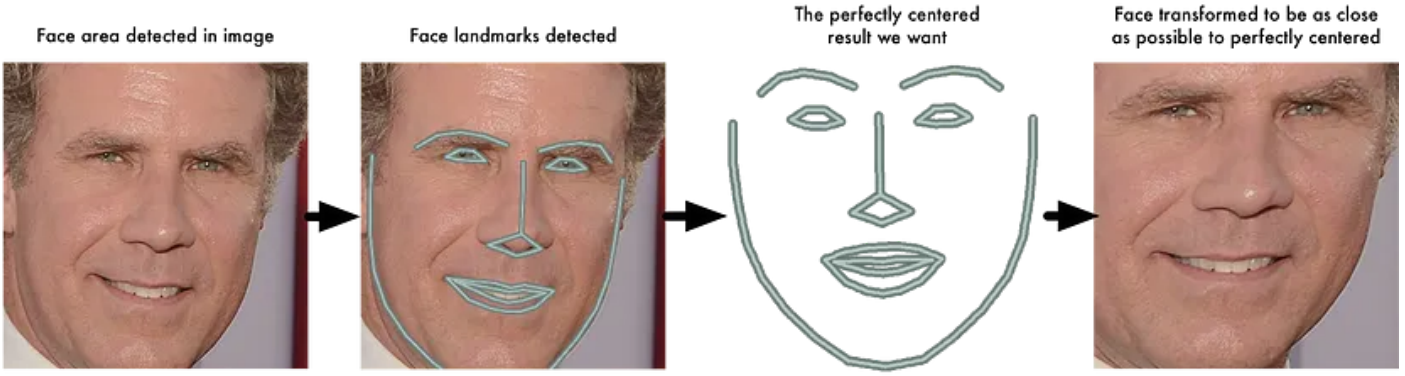
Faces turned in different directions look totally different to a computer:

Humans can easily recognize the same images of a face turned or oriented in a different direction, but computers would see these pictures as two completely different people. To account for this, each picture is wrapped so that the eyes and lips are always in the sample place in the image. This will make it a lot easier to compare faces in the next steps.

To do this, an algorithm called face landmark estimation is used. It is an approach invented in 2014 by Vahid Kazemi and Josephine Sullivan.

The basic idea is to come up with 68 specific points (called landmarks) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then train a machine learning algorithm to be able to find these 68 specific points on any face:

Here's the result of locating the 68 face landmarks on a test image:



Once the eyes and mouth are identified, the image is simply rotated, scaled and sheared so that the eyes and mouth are centered as best as possible. Only basic image transformations like rotation and scale that preserve parallel lines (called affine transformations):

Now no matter how the face is turned, we are able to center the eyes and mouth. And, are in roughly the same position in the image. This makes the next step a lot more accurate.

Feature Extraction using Neural Networks:

FaceNet is a deep learning model that was introduced in 2015 for face recognition. It uses a convolutional neural network (CNN) to extract features from faces and then maps those features into a 128-dimensional space. The FaceNet model uses a triplet loss function to learn the mapping between faces and their corresponding 128-dimensional feature vectors. The triplet loss function compares the distance between an anchor image, a positive image (same person as the anchor), and a negative image (different person from the anchor). The model learns to minimize the distance between the anchor and positive images and maximize the distance between the anchor and negative images.

To extract the 128 features from a face image using the FaceNet model, the input image is first preprocessed to standardize the pixel values and crop the face region. The preprocessed image is then passed through a CNN, which extracts a set of high-level features that capture the unique characteristics of the face. The last layer of the CNN produces a 128-dimensional embedding, which represents the face in a high-dimensional feature space.

The FaceNet model is trained on a large dataset of face images, allowing it to learn a robust representation of faces that is invariant to variations in lighting, pose, and expression. The model can be used for face verification, face identification, and face clustering.

One advantage of the FaceNet model is that it can handle face images with varying poses and expressions, making it useful for real-world applications where faces may be captured under different conditions. Additionally, the 128-dimensional feature vector representation of a face allows for efficient comparison and matching of faces, making it well-suited for large-scale face recognition systems.

Architecture of the model

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Feature classification using Support Vector Machine (SVM)

The Support Vector Machine (SVM) used for classification of the 128 features extracted using the FaceNet model. An SVM is a machine learning algorithm that is commonly used for classification tasks. Given a set of labeled data, an SVM learns a boundary or hyperplane that separates the data into two classes with maximum margin. The margin is defined as the distance between the hyperplane and the closest points from both classes. The SVM aims to find the hyperplane that maximizes this margin, which helps to ensure better generalization to new, unseen data.

In the context of face recognition using the FaceNet model, the SVM is used to classify the 128-dimensional feature vectors extracted from face images. The SVM learns a decision boundary between pairs of feature vectors that belong to the same person (positive pairs) and pairs of feature vectors that belong to different people (negative pairs).

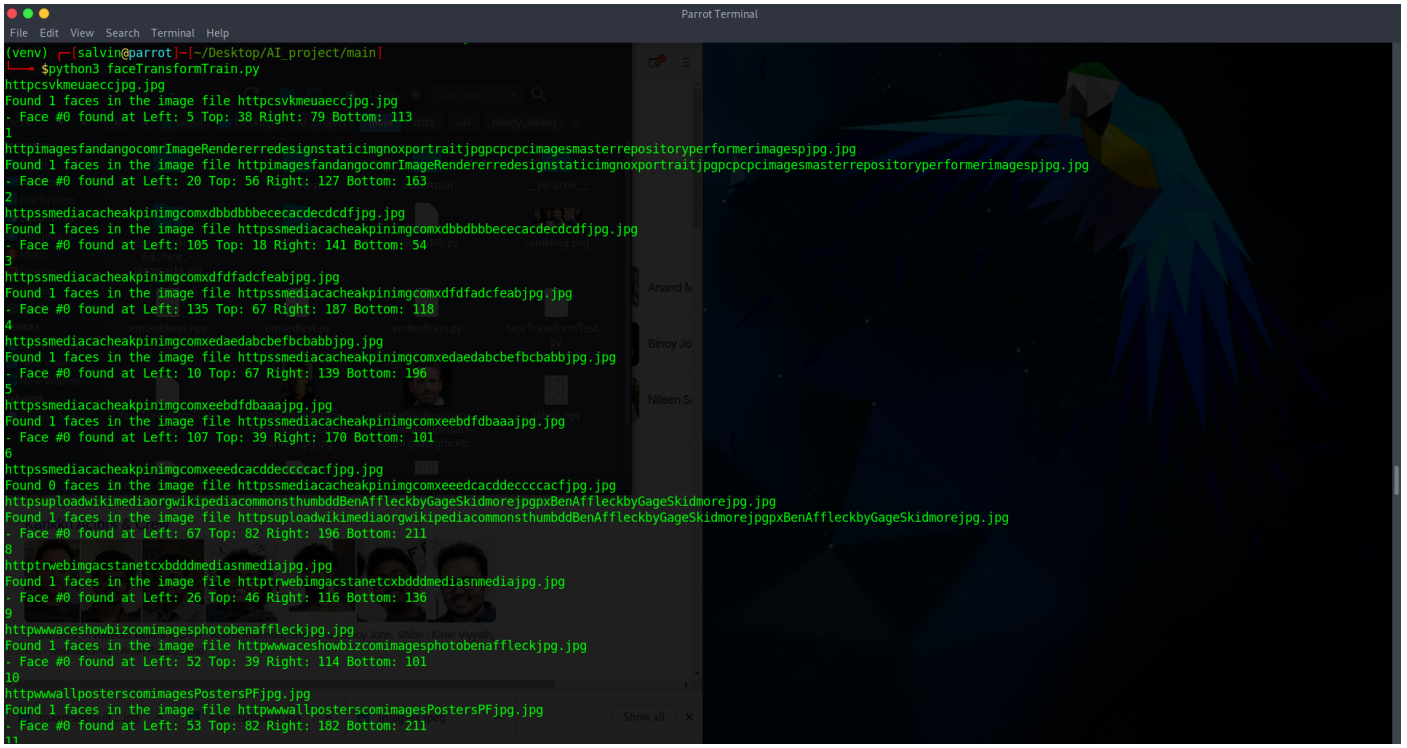
During training, positive and negative pairs of feature vectors are constructed by comparing the feature vectors of all possible pairs of face images in the dataset. The SVM is then trained on these pairs using a binary classification algorithm, where the goal is to learn a decision boundary that separates the positive pairs from the negative pairs with maximum margin.

During testing, the SVM takes a pair of feature vectors and predicts whether they belong to the same person or different people. This prediction is made by evaluating the distance between the two feature vectors and comparing it to a threshold. If the distance is below the threshold, the SVM predicts that the two feature vectors belong to the same person, and if it is above the threshold, the SVM predicts that they belong to different people.

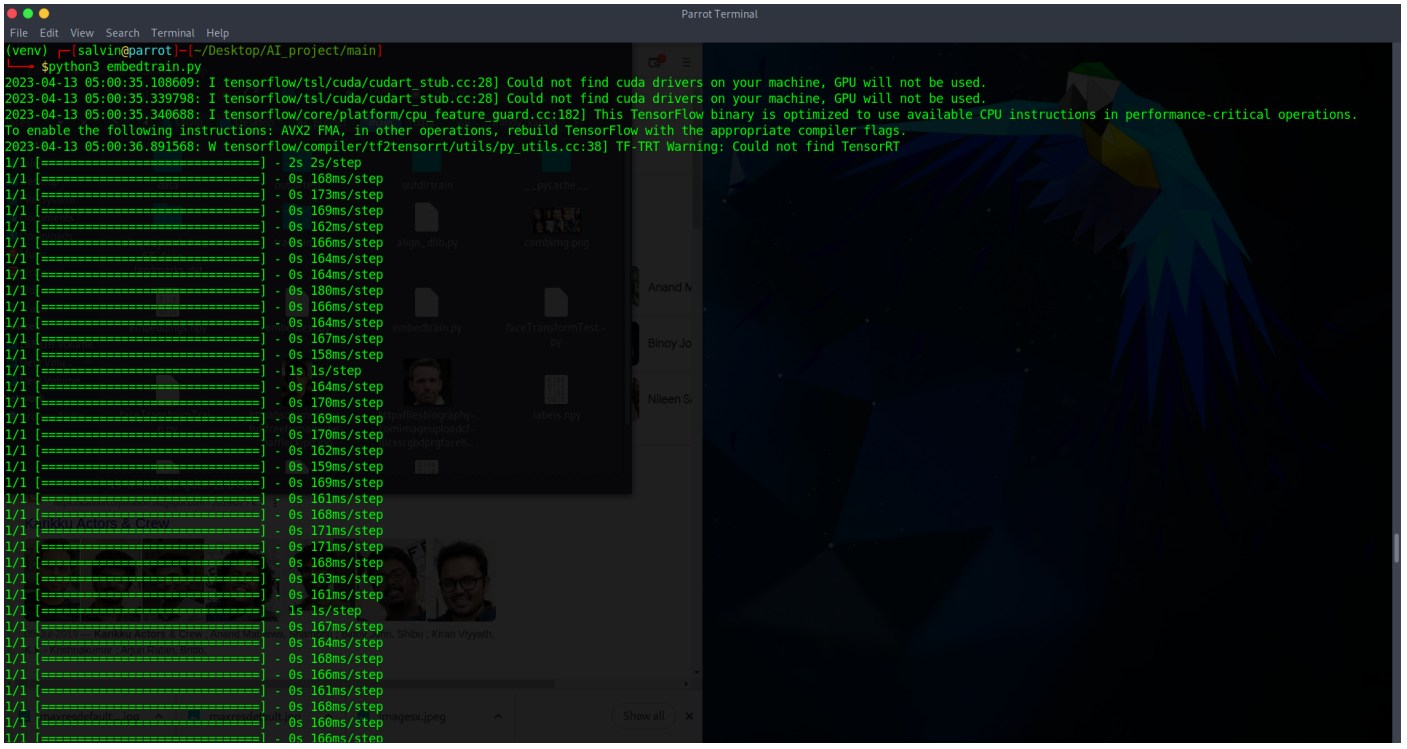
One advantage of using SVM for classification in face recognition is that it can handle high-dimensional data like the 128-dimensional feature vectors produced by the FaceNet model.

Results:

Image preprocessing for training:



Extracting the 128 features from the image for training:

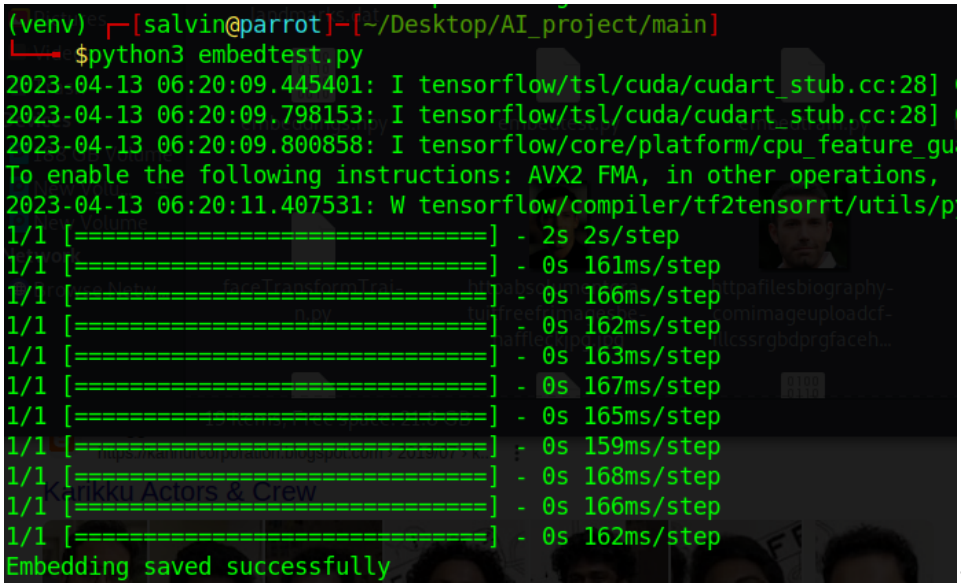


Group photo given for prediction :

Detecting faces in the image and bounding them inside a box

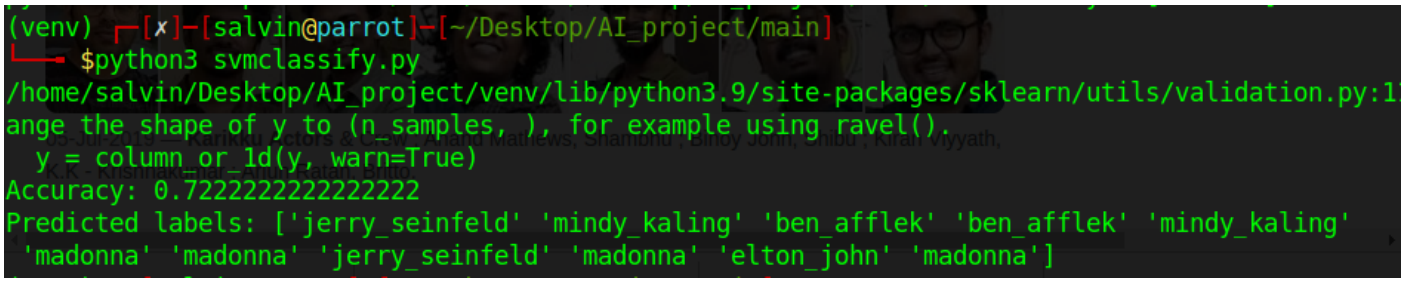
```
(venv) ┌─[salvin@parrot]─[~/Desktop/AI_project/main]
└──╼ $python3 faceTransformTest.py combimg.png
Found 11 faces in the image file combimg.png
- Face #0 found at Left: 46 Top: 374 Right: 201 Bottom: 528
- Face #1 found at Left: 975 Top: 337 Right: 1050 Bottom: 412
- Face #2 found at Left: 469 Top: 98 Right: 655 Bottom: 284
- Face #3 found at Left: 282 Top: 96 Right: 411 Bottom: 225
- Face #4 found at Left: 970 Top: 522 Right: 1022 Bottom: 573
- Face #5 found at Left: 712 Top: 340 Right: 841 Bottom: 469
- Face #6 found at Left: 563 Top: 365 Right: 653 Bottom: 454
- Face #7 found at Left: 982 Top: 66 Right: 1071 Bottom: 156
- Face #8 found at Left: 325 Top: 411 Right: 454 Bottom: 540
- Face #9 found at Left: 700 Top: 98 Right: 855 Bottom: 253
- Face #10 found at Left: 67 Top: 116 Right: 175 Bottom: 223
```

Extracting the 128 features from the image for testing:



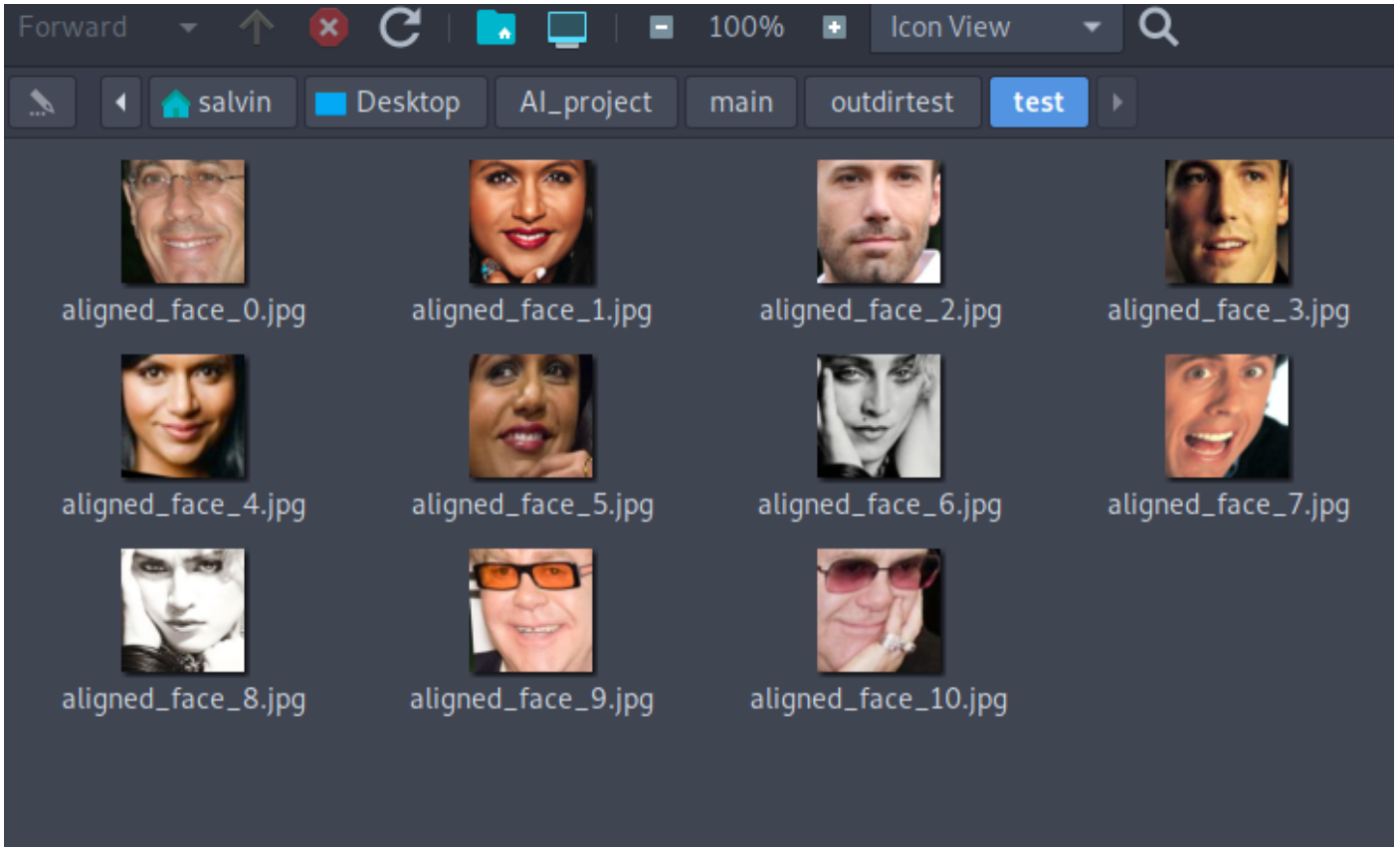
+

Predictions from the model.



The validation accuracy is 0.722.

These are the face detected from the input group photo (in order)



Here except for face_5 and face_10 all other faces were detected correctly.
Hence Accuracy = 9/11 = 81.8%

Conclusion:

In conclusion, the face recognition model described above can be effectively used for automating attendance in a classroom setting. By taking photos of students in a class and passing them through the model, the system can accurately identify each student and mark them as present.

The HOG algorithm is used to simplify the image and identify the parts of the image that look like a generic HOG encoding of a face. This is then used to find the main landmarks in the face and warp the image so that the eyes and mouth are centered. The FaceNet model is then used to extract 128 features from the centered face image, which are used to identify the student. The SVM algorithm is used to classify the extracted feature vectors and determine whether a student is present or not. This system can be highly accurate, especially when used in conjunction with a high-quality camera and proper lighting.

Automating attendance using face recognition can save time and effort for teachers, reduce errors in manual attendance marking, and increase efficiency in the classroom. With further development and refinement, this technology could be implemented in a wide range of educational institutions, from primary schools to universities.

Overall, the face recognition model described in this project can greatly benefit the educational sector and is a promising area for future research and development.

Reference:

1. https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
2. F. Schroff, D. Kalenichenko and J. Philbin, "**FaceNet: A unified embedding for face recognition and clustering**," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.
3. Vahid Kazemi and Josephine Sullivan, "**One Millisecond Face Alignment with an Ensemble of Regression Trees**", KTH, Royal Institute of Technology Computer Vision and Active Perception Lab Teknikringen 14, Stockholm, Sweden {vahidk,sullivan}@csc.kth.se
4. Support Vector Machines, https://scikit-learn.org/stable/modules/svm.html