Facial Recognition System

by

Ashish Kumar Singh

2019BCS-010

A report submitted for Summer Project

Bachelor of Technology

in

Computer Science and Engineering

विश्वजीवनामृतं ज्ञानम्

ATAL BIHARI VAJPAYEE-

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015, MADHYA PRADESH, INDIA

Report Certificate

I hereby certify that the work, which is being presented in the report, entitled **Facial Recognition System**, for Summer Project in **Computer Science and Engineering** and submitted to the institution is an authentic record of my/our own work carried out during the period May-2021 to August-2021 under the supervision of **Supervisor Prof. Karm Veer Arya**. I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date: 10th August 2021

Name and Signature of the candidate

This is certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date: 10th August 2021

Names and Signatures of Research Supervisor

Candidate's Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the check-list and ensure that my thesis is in the prop er format as specified in the guideline for thesis preparation.
I declare that the work containing in this report is my own work. I understand that plagiarism is defined as any one or combination of the following:

(1) To steal and pass off (the ideas or words of another) as one's own

(2) To use (another's production) without crediting the source

(3) To commit literary theft

(4) To present as new and original idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else's work constitute plagiarism.
I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature:

Name: Ashish Kumar Singh

Roll. No: 2019BCS-010

Date: 10th August 2021

Abstract

A Face Recognition System is one of the biometric information processes, its applicability is easier and working range is widespread like fingerprint, iris scanning, signature etc. In my Summer Minor Project, I have first designed and trained a Face Verification System for classifying between two images belonging to the same or different person. After undergoing training for Face Classification our model for Face Verification can also be transformed for identifying the image of a person taken at runtime from a database containing images of many people. The report shows the readers the approach I have taken in order to develop our very own Facial Recognition System. Several famous Face Recognition Algorithms which I have taken into consideration while developing our own Face Recognition System have also been explained thoroughly.

Keywords : Face Verification, Face Recognition, Siamese Network, Deep Neural Network Architecture, Computer Vision

Acknowledgments

I want to extend my heartful thanks to my Supervisor Prof. Karm Veer Arya for constantly guiding me through the project. His in depth knowledge of Computer Vision surely provided me with the right path to approach my problem. Whenever I faced any issue or required any guidance he was always available to talk with me about the same and resovle all my doubts. He even helped me with righting this report.

Table of Contents

Chapter

Tables

Table

List of Figures

Figure

Chapter 1

Introduction

Face Verification System : A Face Verification System is the one in which given two facial images, it tells us whether those two images belong to the same or different person using the facial features extracted from both the images from our trained model(1:1).

Facial Recognition System : A Facial Recognition System has information stored about the facial features of different people in an organization and at run time whenever a new facial image is taken of a person it uses the information about facial features extracted from that image and compares it with the information about facial features of the people which are already stored in the database to predict the identity of that individual(1:K).

1.1 Motivation

COVID-19 : When the unfortunate COVID-19 pandemic stuck India in 2020 it forced many colleges, offices and workplaces to shut down. However, as we now open up and try to adopt to this new normal we still need to follow COVID-19 safety protocols and Social Distancing Practices as prescribed by the government. While we try our best to follow these practices there are certain areas which still can have a scope for improvement. Many offices/colleges (including ours) still use bio-metric attendance for identification which might be not safe during this COVID day and age. Practices like sanitising hands before and after carrying out the bio-metric attendance sure helps but still does not provide complete surety. Replacing bio-metric systems with Facial Recognition at workplaces is a step towards providing that surety.

Opportunities in Computer Vision and Deep Learning : In present times there are an ample amount of opportunities both in jobs and research related field in Computer Vision and Deep Learning. The image identification problem by machines has various applications ranging from security and defence systems, self driving cars, generating machine automated artwork etc. There is an extensive amount of research which has been done in developing new and better algorithms for the above applications but more work needs to be done in developing new algorithms or training the systems which already exist as we gather more and more data in order to achieve human level performance in above mentioned applications.

1.2 Work Flow

Step 1 : Data collection and organisation using the sources available on the internet and distributing that data for training and testing on our Face Verification Model.

Step 2 : Using that data to train a model for performing Face Verification at a random dataset at run-time with utmost accuracy as possible.

Step 3 : Using our Face Verification model to transform it into a Facial Recognition System.

1.3 Report Organisation

The rest of the material is organised as follows:

Chapter 2 : Chapter 2 describes the various papers we have studied and analyzed in order to gain all round knowledge of the subject we are working upon.

Chapter 3 : Chapter 3 describes the approach we have taken for making our project.

Chapter 4 : Chapter 4 highlights our efforts and reports for the same.

Chapter 5 : Chapter 5 describes approaches that could be taken up in future for the application discussed in the report and other similar applications that could be worked upon.

Chapter 2

Literature Review

Facial Recognition in unconstrained images is at the forefront of automating tasks performed by humans. The social and cultural implications of such technologies are far reaching, but the current performance gap between humans and machines in this domain serves as a buffer from having to deal with these implications.

In the 1960s, the earliest Facial Recognition model was built wherein a human had to pinpoint the coordinates of the facial features in a photograph which was used to calculate 20 distances on the face(which would eventually be stored in a database), and at run-time the computer would then compare these distances with different photographs in the database and return the possible match. In 1970s first study was published which involved reading and storing information about the facial features without any human intervention.

Until the 1990s Facial Recognition Systems were developed primarily by using photographic traits of human faces, however, research on Face Recognition to reliably locate a face in the image gained traction with the Principal Component Analysis or PCA. Purely feature based approaches to Facial Recognition were overtaken by this time.

By the early 2000s real-time face detection in video footage became possible. With the rise of the internet and social media in the past 20 years we have large volumes of human facial data now to train existing models or create new ones. While conventional methods of machine learning such as Support Vector Machines and Principal Component Analysis are not able to handle large volumes of data, recent advances in deep learning methods have made it possible to do so.

2.1 Key Related Research

LFW Dataset[1] is a public benchmark for face verificaton/pair matching. The dataset contains more than 13,000 images of faces from the web of 5,749 different individuals. 1680 of the people pictured have two or more distinct photos in the dataset.

Conventional face recognition pipeline consists of four stages: Detect, Align, Represent and Classify. DeepFace[3] revisits both the alignment step and representation step by employing explicit 3D face modelling in order to apply a piecewise affine transformation, and derive a face representation from a nine-layer deep neural network. This deep network consists of both the standard convolution layers and fully connected layers with more than about 120 million parameters. For facial alignment the paper talks about starting with 2D alignment wherein 6 points are detected at the center of eyes, tip of the mouth, nose and other parts of nose. In order to align faces undergoing out of plane rotations a 3D shaped model is used and a 3D affine camera is used to warp the 2D aligned crop to the image plane of the 3D shape which generates a 3D aligned version of the crop. This is achieved by further detecting 67 fiducial points on this 2D aligned crop with their corresponding

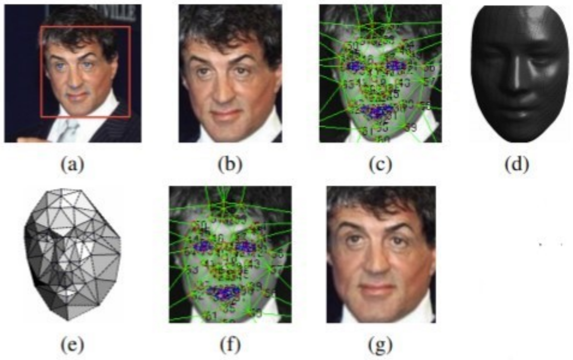


Figure 2.1: Alignment Pipeline[1]

(a) The detected face, with 6 initial fiducial points. (b) The induced 2D-aligned crop. (c) 67 fiducial points on the 2D-aligned crop with their corresponding Delaunay triangulation, triangles are added on the contour to avoid discontinuities. (d) The reference 3D shape transformed to the 2D-aligned crop image-plane. (e) Triangle visibility w.r.t. to the fitted 3D-2D camera; darker triangles are less visible. (f) The 67 fiducial points induced by the 3D model that are used to direct the piece-wise affine warpping. (g) The final frontalized crop.

Delaunay triangulation. The [3] also talks about the Siamese network wherein once learned, our facial recognition network(without the top layer) is called twice for two input images to output whether the two images belong to the same person or not. An absolute difference is taken between the features of both the images followed by top fully connected layer that maps into a single logistic unit. The parameters of the Siamese network are trained by cross entropy loss.

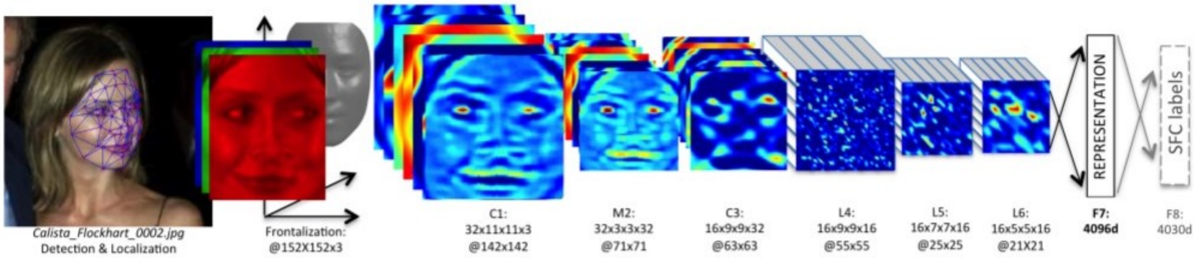


Figure 2.2: Facial Features extraction from an input image[3]

FaceNet[2] sets apart from other methods as it does not require any additional post processing methods unlike in [3]. It does not require a complex 3D alignment but instead only requires a tight crop around face area. A 220 × 220 × 3 array containing RGB values of each pixel of 220 × 220 input image is used to train our Deep Neural Network Architecture to produce image embedding. After that a topmost layer of our model computes whether the two facial input images belong to the same or different person taking those image embedding as an input. Here, a triplet loss function is used to train the topmost layer where the distance between the anchor and the positive(same

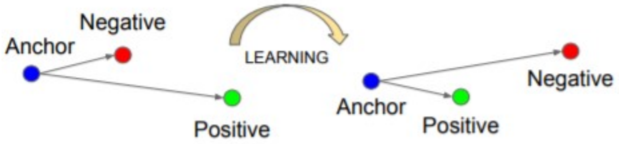


Figure 2.3: Diagrammatic Representation of Triplet Loss[2]

The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity.

identity) is maximized and the distance between the anchor and the negative(different identities) is minimized, such that the distance between anchor and positive is always less than the distance between anchor and negative.

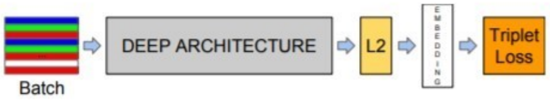


Figure 2.4: Model Structure for training our Face Verification Model[2]

The network in [2] consists of a batch input layer and a deep CNN followed by L2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

2.2 Research Gaps

- In [1] dataset many groups are not well represented. There are very few babies, children, people over the age of 80 and relatively smaller proportion of women. Also people of many ethnic groups are under represented in the dataset. Additional conditions, such as poor lighting, extreme pose, strong occlusions, low resolution, and other important factors do not constitute a major part of [1]. These are important areas of evaluation, especially for algorithms designed to recognize images “in the wild”.
- Networks both in [3] and [2] use a lot of parameters which increases our model size and CPU requirements. Further improvement needs to be carried out in order to reduce our model size and CPU requirements. This also affects our model accuracy as we can train our model only for a limited number of iterations or try out and check for model accuracy by trying out different hyperparameters as training for only one iteration through the entire dataset takes a lot of time.
- Also while training for triplet loss function in [2] anchor, positive and negative needs to be carefully chosen as it generalizes well for most of the examples in our database.

2.3 Objectives

The objective of this project is to first train and develop a Face Verification System and using the weights/parameters/features learned from that Face Verification System to develop a Facial Recognition System.

Chapter 3

Methodology

We would be using a pretrained FaceNet Deep Neural Network Architecture for extracting facial features and generating an image embedding corresponding to the image of a person.

On top of that we would be using MTCNN(Multi-task Cascaded Neural Network) for face detection in a given image and convert it to a desirable size for our FaceNet network.

After that we would be training a binary classifier to identify amongst two image embeddings obtained from our Deep Network, whether they belong to the same or different person.

For our facial recognition system we would be having a database of images of some limited number of people(as is the case in most of the colleges, offices, schools etc. for which this application is purposely being built) and we would be storing the image embeddings corresponding to a given picture of a person. At run time when the image of a person is taken his or her image goes through the computation through our whole network to get their corresponding image embedding. This image embedding is then compared with each embedding in the database and based on the resemblance it has closest to it is used to confirm the identity of a given person.

3.1 Mechanism

Step 1 : Arrange pairs for our training dataset.

Step 2 : Detect faces for each image in our training pair using MTCNN to obtain detected face images and return the corresponding image-arrays according to our desired input(in this case its $160 \times 160 \times 3$) for our FaceNet network.

Step 3 : Normalize our input image arrays according to the formula :

$$x_{ijk} = (x_{ijk} - y)/z \tag{3.1}$$

here,

x_{ijk} is the element corresponding to ith row, jth column and kth channel in x, which is our input image array,

y is the mean for our input image array,

z is the standard deviation for our input image array.

Step 4 : We run our corresponding image arrays through the FaceNet to get a 128 dimensioned image embedding.

Step 5 : Now for each pair we calculate the corresponding absolute difference between its embeddings.

Step 6 : We then train a binary classifier to identify for the two image embeddings belonging to the same or different person whose absolute difference is taken as our input for classifier, wherein we use a Adam optimization method for minimizing our cross entropy loss.

$$z^i = W * x^i + b \tag{3.2}$$

here,

W is a 128 dimensioned weight matrix for our binary classifier(it is randomly initialized in the beginning),

x^i is the absolute difference between image embeddings for ith training pair,

b is our bias weight(it is also randomly initialized in the beginning).

$$y_o^i = e^{z^i}/(1 + e^{z^i}) \tag{3.3}$$

here,

y_o^i is the output predicted by our classifier for whether the images in the training pair belong to the same(1) or different person(0).

$$J^i = y^i * \log(y_o^i) + (1 - y^i) * \log(1 - y_o^i) \tag{3.4}$$

here,

y^i is the actual output for ith training pair(1 if the images belong to the same person and 0 if not),

J^i is the cross-entropy loss for ith training pair.

$$C = \sum_{i=1}^{m} J^i \tag{3.5}$$

here,

C is the total cost, that is the sum of cross entropy losses through all our training examples,

m is the total number of training pairs.

Step 7 : After successfully training our classifier for verification it can be used for the purpose of facial recognition.

Step 8 : We obtain(Step 2 - Step 4) and store the image embeddings for the images of all people our application is supposed to detect.

Step 9 : At runtime whenever an image is taken, it also goes through the same procedure described in steps 2 to 4 to get a 128 dimensioned image embedding.

Step 10 : After that we calculate the absolute difference for the embedding obtained at runtime with respect to every image embedding already stored in the database.

Step 11 : For a difference corresponding to every embedding in our database our classifier returns the value for whether the images corresponding to the two embeddings are of same person(1) or not(0).

Step 12 : The embedding corresponding to the person with respect to which our classifier gives the highest output is confirmed as the identity of the person provided the highest output is greater than 0.5 else none.

3.2 Tools

- **Google Colaboratory** : Google Colaboratory, or “Colab” for short, is a product from Google Research. Google Colab allows anybody to write and execute arbitrary python code through the browser. Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

Documentation : colab.research.google.com

We have run, edit our entire code on Google Colab as it provided much higher computational power as compared to my CPU and that too free of cost. Also, it was possible for us to take photos for our experimentation because of the Google Colab's inbuilt image capturing feature.

- **Python** : Python is an object oriented high-level programming language used for dataset pre-processing and machine learning.

  Documentation : docs.python.org/3

  We have written all our code in Python. Throughout our entire code we have used different libraries of Python for various purposes. From data organization to model training we've carried all these tasks on Python.

- **NumPy** : NumPy is a python library used for working with arrays.

  Documentation : numpy.org/doc

  We've used NumPy for the purpose of storing our RGB pixel values of images in form of arrays while training our model. We've also used NumPy for the purpose of storing image embeddings corresponding to different images in our database.

- **Pandas** : Pandas is a library written for Python for data manipulation and analysis.

  Documentation : pandas.pydata.org/docs

  Pandas was used for storing the image paths of our training and cross-validation dataset images while we were organizing our data for future model training.

- **Keras** : Keras is a minimalist Python library that runs on top of TensorFlow. It provides a Python interface for artificial neural networks. It is mainly used for the purpose of deep learning.

Documentation : keras.io

We used Keras for the purpose of loading pre-trained weights of our Deep Neural Network in [3] and obtaining image embeddings corresponding to our images. We also trained and saved the weights for our logistic classifier using Keras.

- **Python Imaging Library** : Python Imaging Library or PIL is used for handling of different image file formats in Python.

  Documentation : pillow.readthedocs.io/en/stable

  PIL was used for extracting RGB pixel values from the images and vice-versa at various instances in our code.

We have trained a binary classifier for detecting whether the two images belong to the same or different individual. After successfully training our model for Face Verification we are ready to use it for Face Recognition purposes. We have created a database of images of different people and stored their corresponding image embeddings. At runtime when a photograph of a person is taken, embedding corresponding to that person is generated. That embedding is compared with the embedding of every person in our database to confirm the identity of the person.

Chapter 4

Experiments and Results

4.1 Experiment Design

4.1.1 Parameter Settings

- We have chosen 3600 pairs of images from the Dataset in [1] for the purpose of training our binary classifier.
- We have chosen 1200 pairs of images from the Dataset in [1](which do not coincide with any of the training pairs)

For training our binary classifier absolute difference between the embeddings of training image pairs is taken as input.

- We have decided our batch size for learning weights to be 600. The input is randomly shuffled after every epoch so that our binary classifier is not subject to any kind of generalization.
- We have fixed number of epochs that is total number of iterations through our entire training dataset for learning to be 1000.

As described in the Methadology we have used Adam Optimization method for learning our weights and minimizing our cost,

$$m_t = b_1 * m_{t-1} + (1 - b_1) * \frac{\partial L}{\partial W_t} \tag{4.1}$$

here,

m_t is aggregate of gradients at time t, initially m_t is 0,

m_{t-1} is aggregate of gradients at time t-1,

∂L is the derivative of loss function(described in equation 3.4 Chapter 3),

∂W_t is the derivative of weights for our binary classifier at time t.

$$v_t = b_2 * v_{t-1} + (1 - b_2) * [\frac{\partial L}{\partial W_t}]^2 \tag{4.2}$$

here,

v_t is sum of square of past gradients, initially v_t is 0,

∂L is the derivative of loss function(described in equation 3.4 Chapter 3),

∂W_t is the derivative of weights for our binary classifier at time t.

$$m_t^c = m_t/(1 - b_1^t) \tag{4.3}$$

$$v_t^c = v_t/(1 - b_2^t) \tag{4.4}$$

$$w_{t+1} = w_t - m_t^c * a/(\sqrt[2]{v_t^c} + e) \tag{4.5}$$

In the above set of equations(4.1-4.6), we have hyper-parameters, b_1, b_2, a, e.

We have set our hyper-parameters as,

- $a = 0.001$
- $b_1 = 0.9$
- $b_2 = 0.999$
- $e =$ 1e-07

4.2 Results

The loss is calculated by carrying out the summation of cross-entropy loss for every input example(Refer to Equation 3.5 in Chapter 3). The accuracy is calculated using the formula :

$$acc = (cp/tp) * 100 \tag{4.6}$$

here,

acc is the accuracy

cp is the number of correct predictions

tp is the total number of predictions

Our prediction is correct if for a training example say the output is 1, then in that case our classifier should give an output greater than or equal to 0.5. In case the output is 0, then in that case our classifier should return a value less than 0.5.

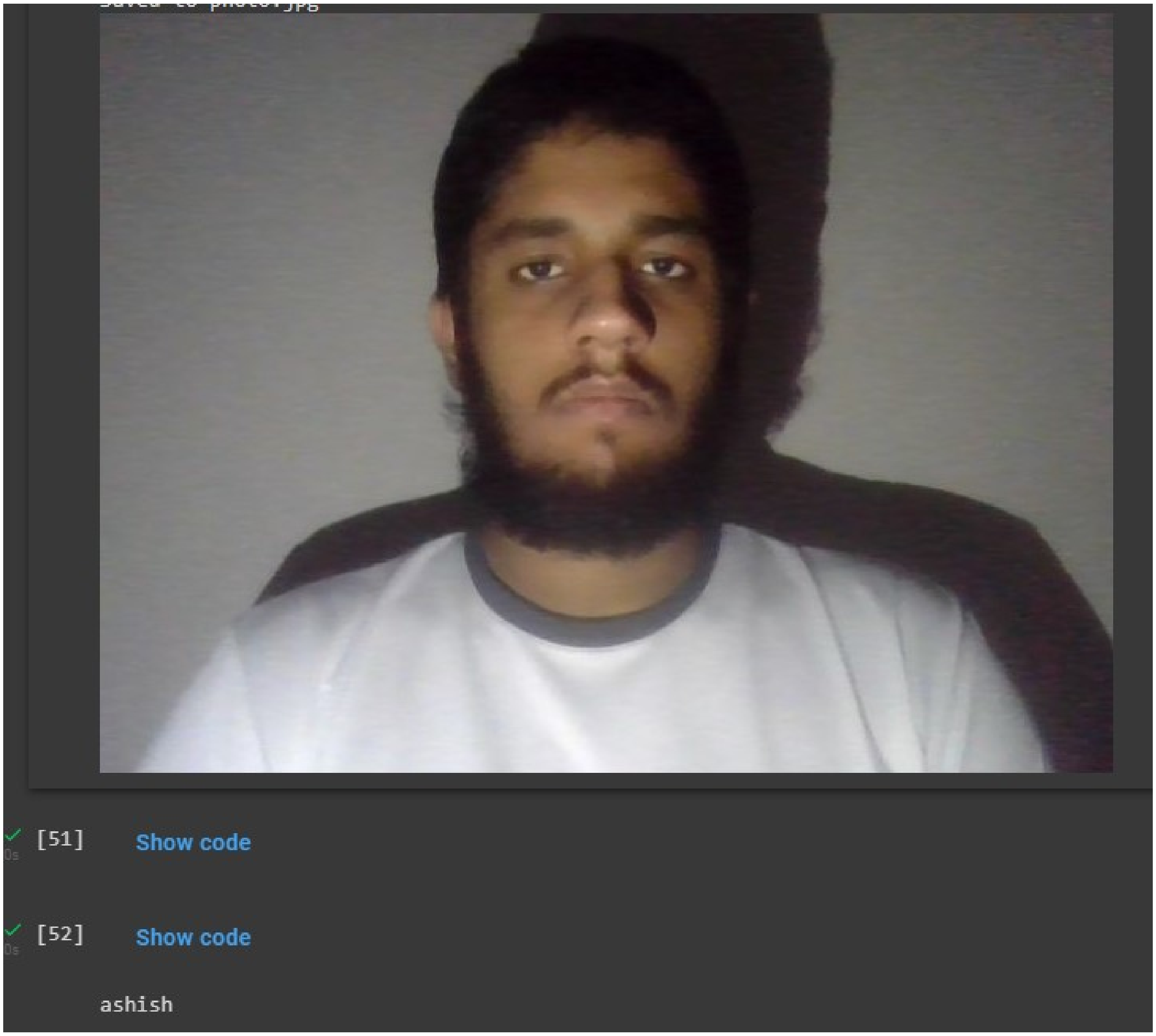
Table 4.1: Performance of our Binary Classifier on the Training Dataset

epochs	loss	accuracy
1	0.3643	92.69
10	0.3611	92.83
25	0.3560	93.22
50	0.3475	93.56
100	0.3314	93.97
200	0.3027	94.83
300	0.2780	95.53
400	0.2566	95.81
500	0.2379	95.97
600	0.2219	96.06
700	0.2077	96.17
800	0.1955	96.31
900	0.1848	96.39
1000	0.1757	96.44

The Table 4.1 shows the performance of our classifier on our entire training dataset. As you can see, as the number of epochs is increasing our classifier is getting more and more optimized to report for two image embeddings. Our loss is decreasing with an increase in passes through our dataset with an increase in accuracy.

Table 4.2: Performance of our Binary Classifier on the Cross Validation Dataset

loss	accuracy
0.1920	95.92



```
saved to photo.jpg
```

```
ashish
```

Figure 4.1: Experimental Run of our Facial Recognition System

Our Facial Recognition System is able to identify me out of a database containing images of 26 people.

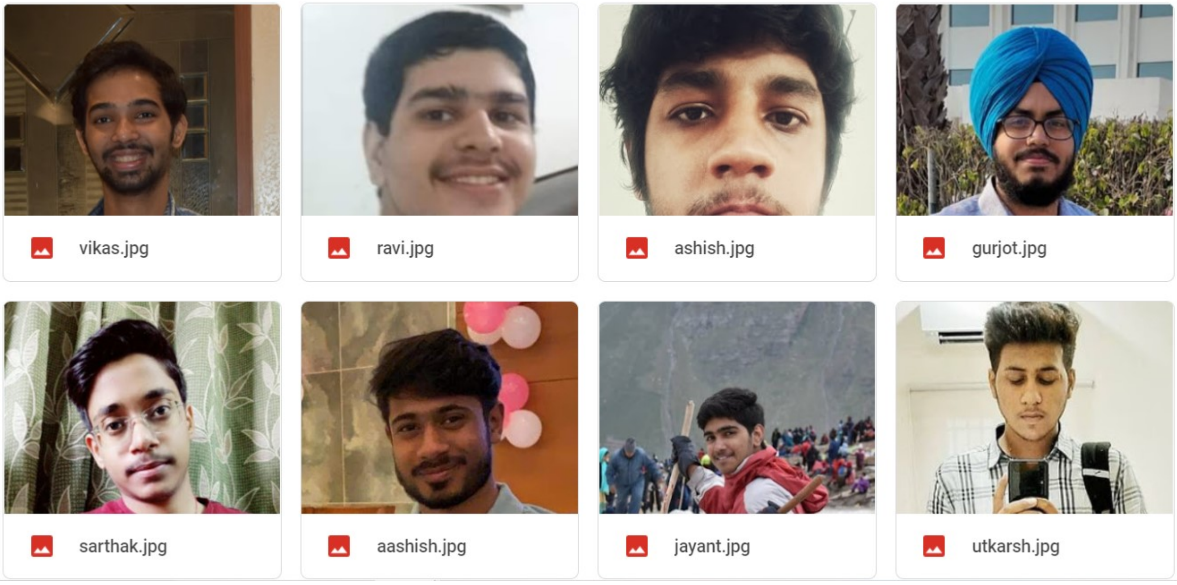


Figure 4.2: Database for our Facial Recognition System

4.3 Conclusion

We've got promising results as we expected. The accuracy of our classifier for our Face Verification could be further increased if we train it for more number of epochs ensuring that our classifier does not overfit for our dataset. We can also train our classifier for different datasets apart from the dataset in [1]. Also, similarly our FaceNet Deep Neural Network architecture could also be further trained on different and even bigger datasets provided we have ample amount of computational resources. All this further training could help us in improving the performance of our Facial Recognition System as a whole and help us approach much and much closer towards human level performance.

Chapter 5

Conclusion and Future Scope

The work presented in this report is a step forward to a new way of thinking in Face Verification and Face Recognition. The novel approach taken in this model of combining ideas from two research papers : [3] and [2], will inspire upcoming researchers to improve performance in different machine learning tasks by taking inspiration from previously carried out research and combining the ideas discussed in them. The model in its current state is able to recognize faces which belong to our database. The model is good enough and could be used for the purpose of attendance and for security at work places, offices, colleges, schools etc.

5.1 Limitations

We don't have a liveness detection feature in our system. One may easily fool the cameras with an image of another person, information about whom is stored in the database to breach through our system.

Also, even though our system does perform well in low light/dim light but in case of absolute darkness which is possible in real life scenario our system fails to deliver the way a human might be expected to.

Also, for training and developing the entire system by ourselves from the beginning through our entirely own and unique approach is not possible due to the economic constraint as it requires very high computational resources.

5.2 Future Scope

As described above that our project lacks a liveness detection feature. Work could be carried out on liveness detection with facial recognition system.

For the purpose of facial recognition we have used the concept of Siamese Network wherein image embeddings obtained for a person at runtime is compared with all image embeddings in the system. Even though this approach has its advantages work should also be carried out on developing a deep neural network wherein the output layer can be increased or decreased according to the number of people which keep changing in our database without adding much expense on our computational costs.

Bibliography

[1] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[2] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 815–823, 2015.

[3] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 1701–1708, 2014.