AUTOMATIC FACE RECOGNITION BASED ATTENDANCE SYSTEM FOR TEACHING LARGE CLASSES

A project report,

submitted to **Dr. Sunil Kumar**

in the subject of Machine Learning

by

GROUP-4

Amal Shaji (2017BCS-010)

Pobbati Samanyu (2017BCS-021)

Pradyumn Pottapatri (2017BCS-022)

Sandarbh Yadav (2017BCS-027)

Vaibhav Garg (2017BCS-038)

Vikram Chaudhary (2017BCS-039)

विश्वजीवनामृतं ज्ञानम्

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT, GWALIOR-474 015

ABSTRACT

Attendance is an important component in modern education system. Currently, traditional methods are used for taking attendance in majority of educational institutes. These include calling out names or roll numbers, passing the attendance sheet etc. These methods waste a lot of time of the lecturer especially when the classes are large enough. There is a need of an automated system for taking attendance. Automatic face recognition based attendance system is a system whose purpose is to automate the attendance process in large classes. It uses Haar cascade classifiers for face detection and one shot learning for face recognition. Haar cascade classifiers are one of the best algorithms for face detection. For face recognition, use of Deep Convolutional Neural Networks is a very popular approach. But this approach has its own limitations. A lot of training data is required and the method is not robust in itself. In order to overcome these limitations, we have used one shot learning. It tries to learn an image by looking at only once. This eliminates the requirement of huge amount of training data. The proposed system uses Contrastive Loss as the loss function and delivers pretty good performance on test data.

Keywords: Face detection, Face recognition, One shot learning, Contrastive loss, Haar cascade classifier etc.

ACKNOWLEDGEMENTS

We are very thankful to **Dr. Sunil Kumar** who provided us the liberty to experiment various ideas. We are highly indebted to him for guiding us and showing high amount of interest in our work. He motivated us and boosted our confidence. This proved beneficial for us and infused trust within us. This project has been completed only because of his precious suggestions, intelligent judgement and fair criticism. He resolved our each and every doubt and we never felt as beginners. He listened to our opinions and appreciated and improved them. The only reason our work has been completed is due to his constant support and helpful attitude.
Lastly, we thank our companions and the institute for their continuously uplifting which helped in renewing our spirits, refocus our attention and helping us in completing this project.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
FD	Face Detection
FLD	Fisher's Linear Discriminant
FLOPS	Floating point Operations Per Second
FR	Face Recognition
GB	Giga Bytes
LBP	Local Binary Pattern
LFW	Labeled Faces in the Wild
MB	Mega Bytes
MKD	Multi Keypoint Descriptor
ML	Machine Learning
NN	Neural Networks
OSL	One Shot Learning
PCA	Principal Component Analysis
QR	Quick Response
RFID	Radio Frequency Identification
SRC	Sparse Representation-based Classification
TPU	Tensor Processing Unit

CHAPTER 1

Introduction

Attendance is an important component in modern education system. Currently, traditional methods are used for taking attendance in majority of educational institutes. They consume a lot of precious time of the lecturer especially when the classes are large enough. The proposed method uses face recognition approach to mark the attendance of students. There are 2 important phases in this approach: Face Detection and Face Recognition. Face detection refers to the process of detecting faces from a given image. This involves separation of faces from non-faces and then cropping the image to extract an array of faces. For face detection, Haar cascade classifiers are the state-of-the-art. Deep Convolutional Neural Networks have played a major role in many image based machine learning or deep learning tasks. It has become state-of-the-art in classification tasks. From image classification to instance segmentation, deep learning hasn't stopped to amaze. There are CNN that could identify 1000 different classes. One such classification task is face recognition. Face recognition is nothing but classifying an image into known label. But one problem with deep CNNs is the requirement of large amount of training data. In many applications, like face recognition, collecting such amount of data is not possible. This is also not possible in an institution where intake is very frequent. For this we need to retrain the model every now and then. This is where one shot learning comes into play. Originally inspired from signature verification, one shot learning tries to learn an image by looking at it once (or with a single training image). One shot learning was initially proposed in FaceNet [1]. Same technology can be seen in action in Baidu headquarters, where it is used for employee attendance system. The model we created is inspired from Facenet. We have tried to minimize the training complexity and model size.

1.1 Motivation

The motivation for the proposed system came by observing the limitations of traditional methods of taking attendance. There are different ways of taking attendance in traditional way: calling out names or roll numbers, passing the attendance sheet etc. But these methods are time consuming and there is high possibilty of false attendance. Also, a lot of manual labour is involved in calculating attendance percentage of various students at the end of the semester. Some alternatives to traditional methods include use of RFID tags, bar codes, QR codes, biometrics etc. But these approaches have their own limitations too. There is high possibility of false attendance in case of RFID tags. One student can handover his RFID tag to his friend to get his attendance marked. Barcodes and QR codes suffer from same limitation. The main issue with biometrics like fingerprint scan is that they involve contact. This is a major limitation especially in today's world suffering from Corona pandemic. Other biometrics like retina scan or face scan are very costly and it is not feasible to install them outside every classroom. Also the attendance taking process can be time consuming in case of large classes as attendance can be marked one by one only. So, we thought of automating the method of taking attendance by using face recognition approach.

Figure 1.1: Queues to mark biometric attendance

1.2 Problem Statement Formulation

Until the introduction of FaceNet [1], all facial recognition system implemented a classification model. If a college course has 'N' students, then the training dataset would contain 'N' classes with each class containing upto 30-40 training images. Then a simple image classification model is trained on the data. Such model has multiple drawbacks:

- Large number of images per class is required
- If the model is trained for 'N' classes, it becomes obsolete when a new student joins. The model has to be retrained again.

There are applications where we neither have enough data for each class and the total number of classes is huge as well as dynamically changing. Thus, the cost of data collection and periodical re-training is too high. Our aim is to develop a model which overcomes the limitations of previous models. This is achieved by using one shot learning which tries to learn an image by looking at it once. In one shot learning, we require only one training example for each class.

1.3 Features

The important features of the proposed system are mentioned below:

1. Physical interaction from users is not required. This is an advantage over fingerprint scan approach which requires contact between user and the fingerprint scanner.
2. There is no chance of false attendance. This is an advantage over RFID, bar code or QR code based attendance approaches.
3. Wastage of time and manual labour is significantly reduced. This is a major advantage over traditional attendance taking methods which are still used in colleges all over the country.
4. The system has an advantage of ubiquity and universality over other biometrics. Everyone has a face and readily displays it.
5. Any good quality camera can be used to capture the data.
6. It is environment friendly as it eliminates the use of paper. This is an advantage over traditional attendance taking methods in which attendance sheets made of paper are used.

1.4 Report Layout

The report is broadly divided into 8 chapters. Chapter 1 provides introduction to the project. It also discusses motivation for the project, problem statement formulation and features of the project. Chapter 2 discusses the literature survey done for the project. Tools and technologies used are given in chapter 3. System design forms the content of chapter 4. Chapter 5 gives detailed overview of the methods used in the project. Chapter 6 discusses the working of the project. Results are provided in Chapter 7. Conclusion and future work is discussed in Chapter 8. References are given at the end.

CHAPTER 2

Literature Survey

Attendance management systems using facial recognition techniques have evolved tremendously in the last decade. Many researchers have used different techniques for face detection and face recognition. Some popular face detection techniques include Viola-Jones algorithm, Haar cascade classifier, colour based techniques, skin classification methods etc. Some popular face recognition techniques include PCA, two-dimensional Fisher's linear discriminant (2DFLD), eigenfaces, correlation methods, Local Binary Pattern(LBP), 3D modelling etc.

Abhishek Jha [2] used colour based technique for face detection. This method detects the colour of humans and its variations. The skin area is then segmented and fed as an input to the recognition process. For face recognition and feature extraction, PCA is used. However, there are many limitations to this approach. The skin tones vary dramatically within and across individuals. Also, due to changes in ambient light and shadows, the apparent colour of the image changes. The movement of persons also causes blurring of colours.

Shehu et al [3] used real-time face detection algorithm. Their approach uses a digital camera installed in a classroom scanning the room every 5 minutes to capture the images of the students. Haar classifier is used for face detection. For face recognition, eigenface methodology is implemented. However, the students are required to pay attention to the camera while capturing images. Their method detects objects as faces creating a large number of false positives. A drastic change in the student's appearance causes false recognition of the student.

Surekha et al [4] used Viola–Jones algorithm for face detection. A sub-window is swept across the selected real-time image for catching the faces. The locator is operated every time with a different size through one image at a time. It uses AdaBoost algorithm to select important features. It employs the MKD-SRC method of partial face recognition irrespective of whether the face detected is holistic or partial. This approach represents database images and real-time images as multi keypoint descriptor (MKD) and then applies sparse representation-based classification (SRC) for face recognition.

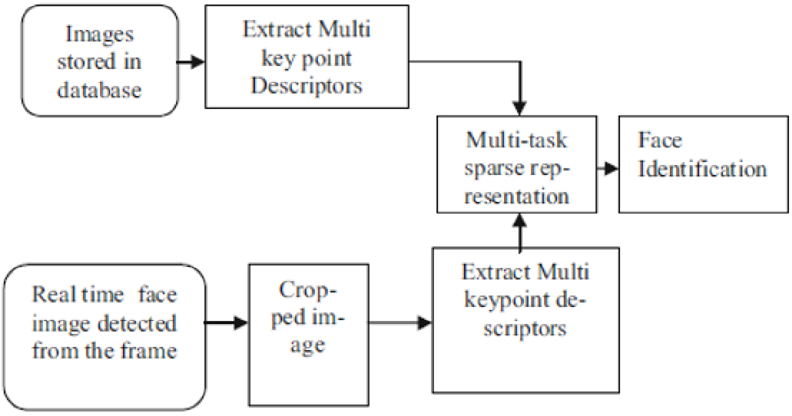

Figure 2.1: MKD-SRC method for partial face recognition

The face detection and face recognition accuracies of some approaches are given below:

Author	Face Detection	Face Recognition	FD Accuracy	FR Accuracy
Muhammad et al	Haar cascade	Eigenface	50%	30%
Naveed et al	Viola Jones	Eigenface	75%	63%
Rekha et al	Viola Jones	Correlation	70%	90%
Mrunmayee et al	Viola Jones	LBP	78%	83%
Surekha et al	Viola Jones	MKD-SRC	80%	60%

2.1 FaceNet: A Unified Embedding for Face Recognition and Clustering

This is the base paper used for our project. It was published by Google researchers Florian Schroff, Dmitry Kalenichenko and James Philbin in 2015. They proposed a system named FaceNet which can implement face recognition on a large scale. FaceNet learns a mapping from the images of faces to a space where Euclidean distance can be used to measure similarity between faces. FaceNet uses CNN as its deep learning architecture and triplet loss as the loss function.

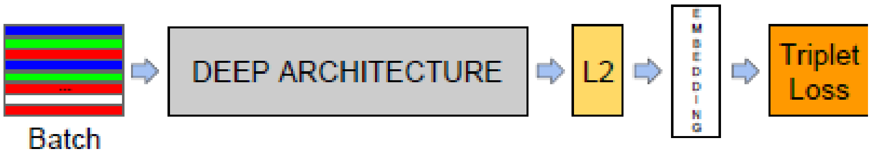

Figure 2.2: Model structure of FaceNet

Triplet loss takes three images for comparison and does not take in any class labels. The objective of triplet loss is to minimize the distance between the similar images and maximize the distance between dissimilar images. The distance between anchor and positive image is minimized and the distance between anchor and negative image is maximized.

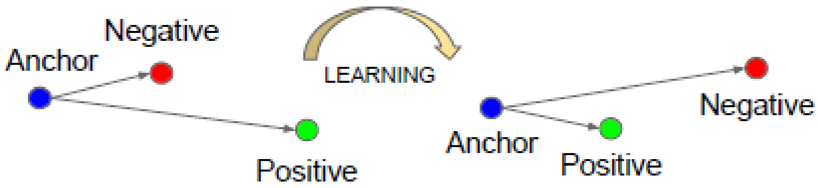

Figure 2.3: Learning using triplet loss

FaceNet contains a lot of parameters (approx 140 million) and is trained on multiple TPUs for several weeks. It performed about 1.6 billion floating point operations per second (commonly known as FLOPS). The size of trained model of FaceNet is in gigabytes(GB). It has a record breaking accuracy of 99.63% on Labeled Faces in the Wild (LFW) dataset. It is 95.12% accurate on YouTube Faces dataset. The model we created is inspired from Facenet. We have tried to minimize the training complexity and model size.

CHAPTER 3

Tools and Technologies Used

The following tools and technologies are used in the proposed system:

3.1 Python

Python is a programming language which is known for its simplicity and widespread adoption. The whole project has been written in Python.

Figure 3.1: Python

3.2 PyTorch

There are many open source libraries for machine learning and deep learning related tasks. The most popular ones are Keras, TensorFlow and PyTorch. There is an ongoing debate in the computer science community to choose between PyTorch and TensorFlow. Although both PyTorch and TensorFlow are fast, we chose PyTorch simply because we have used it before and are familiar with it. PyTorch is also better in code readability and debugging. Most importantly, PyTorch is more pythonic. It feels like using any other library.

Figure 3.2: TensorFlow vs PyTorch

3.3 OpenCV

OpenCV is an open source library which is used for computer vision related tasks. We have used OpenCV for detecting faces in an image and to crop the image to faces efficiently.

Figure 3.3: OpenCV

3.4 Python Imaging Library, Numpy and Matplotlib

Python Imaging Library is used to read images while creating DataLoader. Numpy is used to do image formatting and to convert final tensor predictions. Matplotlib is used for plotting graphs.

CHAPTER 4

System Design

Figure 4.1 gives an overview of the proposed system. First of all, a database of student's personal information (like name and roll number) along with the images of their faces is created. As the proposed system uses one shot learning, one image of each student is sufficient. The image of the students attending class is attained by a camera. Appropriate preprocessing techniques are applied before doing face detection. Haar classifier is used for face detection. The face detection phase gives an array of extracted faces as output. Now, face recognition is done on each of the extracted face by comparing with the images stored in the already created database. Finally, attendance is marked.

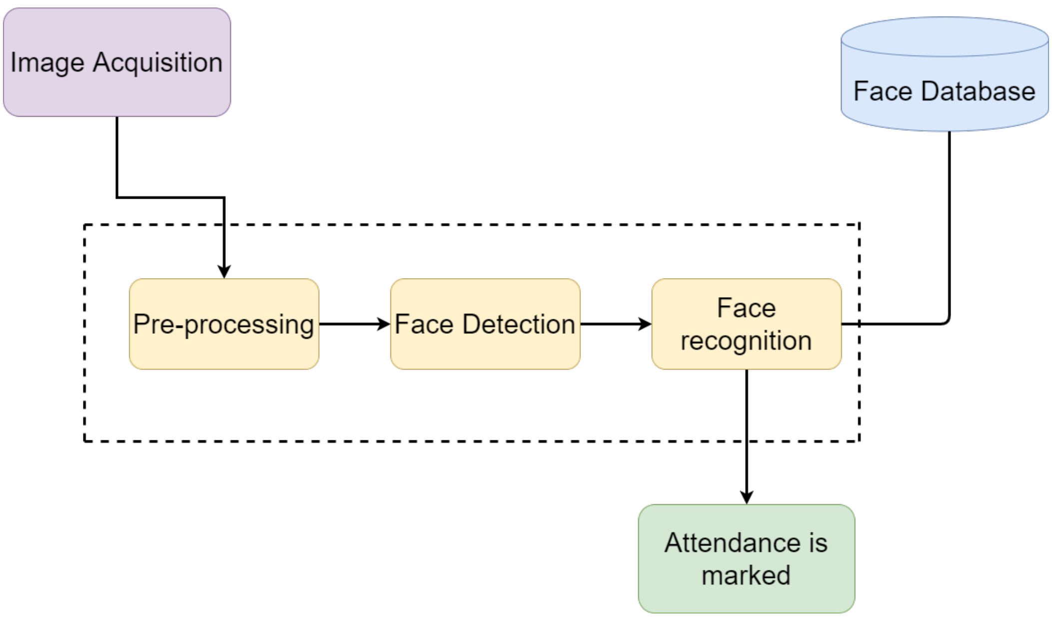

Figure 4.1: System Design

CHAPTER 5

Methodology

5.1 Haar Cascade Classifier

Haar cascade classifier is used to identify objects based on the concept of features as proposed by Viola-Jones algorithm. In this algorithm, the cascade function is trained from a large number of positive and negative images. The classifier is then used to detect the object in other images. For the face detection process, the algorithm requires an input stream of positive (with faces) and negative images (without faces) to train the classifier. Features are extracted from the training set. Haar like features are collected which contain adjacent rectangular regions at a specific location in a detection window. The pixel intensities are summed up in each region and the difference between these sums is calculated. Then, consecutive feature scaling and rectangular scaling is done. When a test data set is brought, the algorithm uses the cascade classifier to detect the faces.

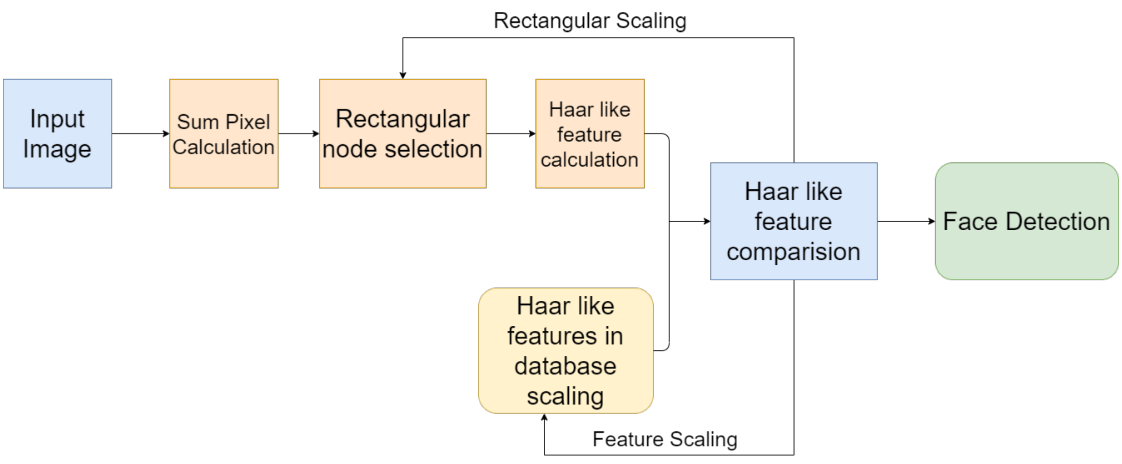

Figure 5.1: Face Detection using Haar Cascade

5.2 Siamese Network

One shot learning is done with the help of Siamese Network. Figure 5.2 shows the basic block of Siamese Network.

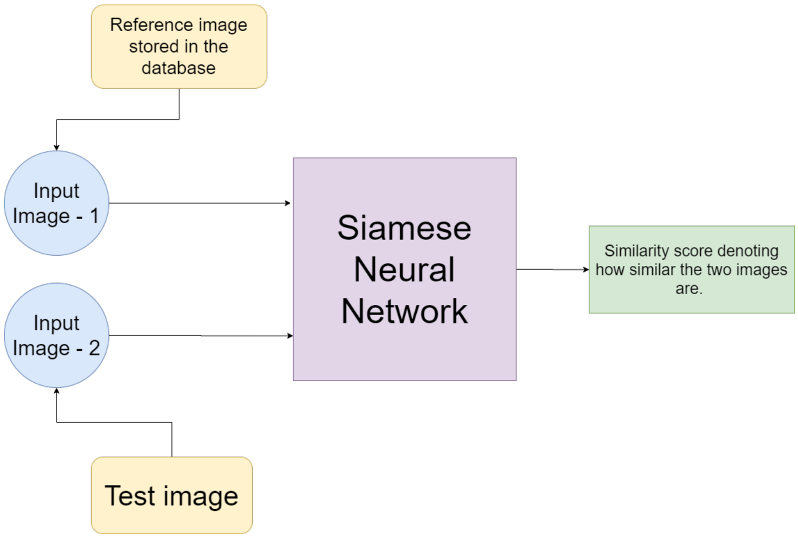

Figure 5.2: Basic block of Siamese Network

Instead of taking an image as input and predicting a class, a siamese network takes two images as input and produces an output which tells how similar the images are. Figure 5.3 shows the architecture of Siamese Network. Usually, the similarity score is passed through a sigmoid function to scale it in the range [0, 1]. A similarity score of 1 means that the images are exactly same while a similarity score of 0 means that the images are completely dissimilar.

The proposed model works by telling how close two input images are by measuring the distance between them. Classification tasks using deep learning optimize the weights of the layers to make a prediction, but here we directly optimize the encoding which is an alternate representation of the image. The encodings are passed through a function that calculates the Euclidean distance between them. If the distance between the encodings is very small, the images are similar and if the distance between the encodings is large, the images are dissimilar.

So in simple terms, the model is not learning to distinguish between images, but is learning to create 'n' length vectors(encodings) that can better decribe an image. The model can then work on other images which it hasn't seen before, because it's not learning to recognize images, but to create encodings. In technical terms, the model is kind of lazy learner. Figure 5.4 illustrates this.

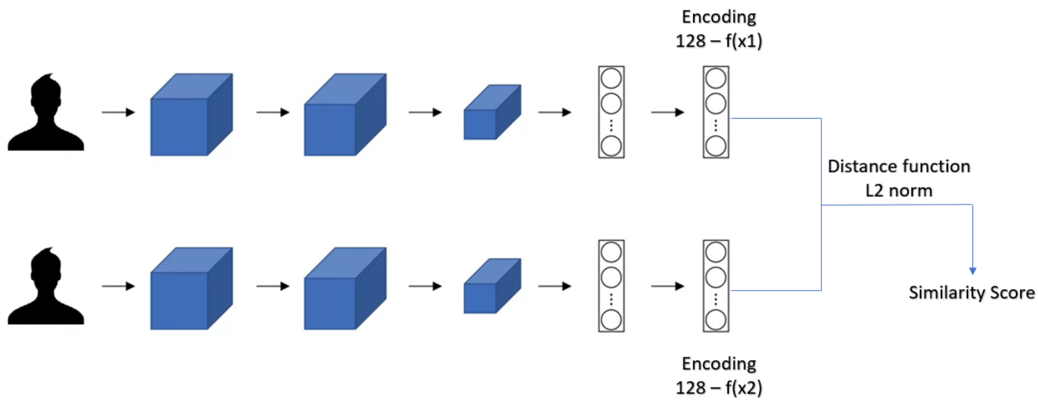

Figure 5.3: Siamese Network Architecture

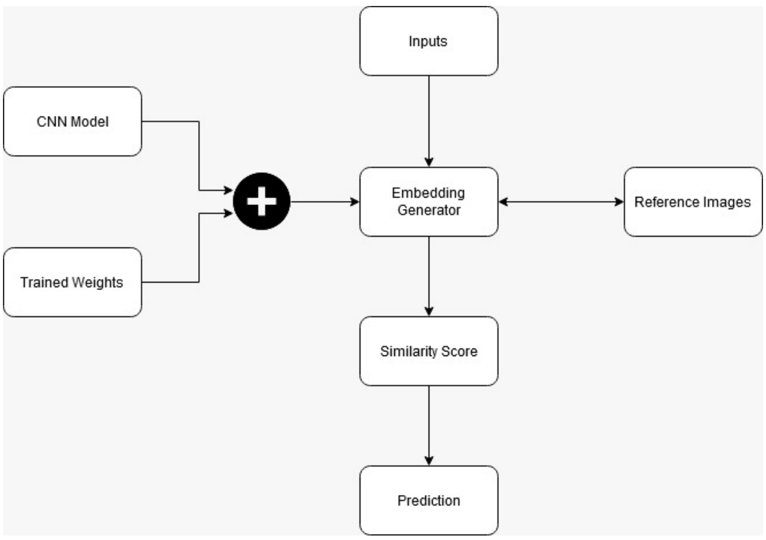

Figure 5.4: Face Recognition using One Shot Learning

5.3 Loss Functions

We consider two loss functions:

- Contrastive Loss [1] [5]
- Triplet Loss [6]

5.3.1 Contrastive Loss

Contrastive loss function takes in two images at a time along with a value describing if they belong to the same class, 0 for same class and 1 for different. So each input takes three values: two images and a class similarity label. The exact loss function is:

$$L(W, L, X_1, X_2) = (1 - Y)\frac{1}{2}(D_w)^2 + (Y)\frac{1}{2}max(0, m - D_w)^2$$

- D_w is the Euclidean distance between encodings of two images (X_1 and X_2)
- Y is the label: 0 for same class and 1 otherwise
- m is the margin parameter. It acts as a radius, the dissimilar pairs only contribute to the loss, if they're within the margin.

5.3.2 Triplet Loss

Triplet loss is the most used loss function in recognition related tasks. It takes three images for comparison and does not take in any class labels. The objective of triplet loss is to minimize the distance between similar images and maximize the distance between dissimilar images. We aim to minimize distance between anchor and positive image and maximize the distance between anchor and negative image . The exact loss function is:

$$L(A, P, N) = max(d(a, p)^2 - d(a, n)^2 + m, 0)$$

- d(a, p): distance between anchor and positive image
- d(a, n): distance between anchor and negative image
- m : margin parameter. It is to ensure that the predictions are confident. For example, if the threshold is 0.51 and the model gives a prediction of 0.50, even though the prediction is true (0.50 < 0.51), but the confidence is poor. So we add a margin score of 0.2 and the prediction holds false (0.7 > 0.51). The dafault margin score used in contrastive loss is 2.0 and in triplet loss is 0.2 (from FaceNet paper).

CHAPTER 6

Working

6.1 Model Structure

Input Shape: [38, 1, 100, 100]

Layer (Type)	Output Shape	No: of Parameters	No: of Trainable Parameters
ReflectionPad2d	[38, 1, 102, 102]	0	0
Conv2d	[38, 4, 100, 100]	40	40
BatchNorm2d	[38, 4, 100, 100]	8	8
MaxPool2d	[38, 4, 34, 34]	0	0
ReflectionPad2d	[38, 4, 36, 36]	0	0
Conv2d	[38, 8, 34, 34]	296	296
BatchNorm2d	[38, 8, 34, 34]	16	16
MaxPool2d	[38, 8, 12, 12]	0	0
ReflectionPad2d	[38, 8, 14, 14]	0	0
Conv2d	[38, 8, 12, 12]	584	584
BatchNorm2d	[38, 8, 12, 12]	16	16
MaxPool2d	[38, 4, 4, 4]	0	0
Linear	[38, 500]	64,500	64,500
Linear	[38, 500]	250,500	250,500
Linear	[38, 128]	64,128	64,128

Total Parameters: 380,088

Total Trainable Parameters: 380,088

Total Non-Trainable Parameters: 0

6.2 Training

The model is trained for 500 epochs because we are getting good results after 500 epochs. Each epoch consists of 10 iterations. The model is trained on a dataset containing images of students from our class. When the model is tested against random images from the internet, it is able to recognize multiple faces successfully.

Figure 6.1: Sample batch for contrastive loss

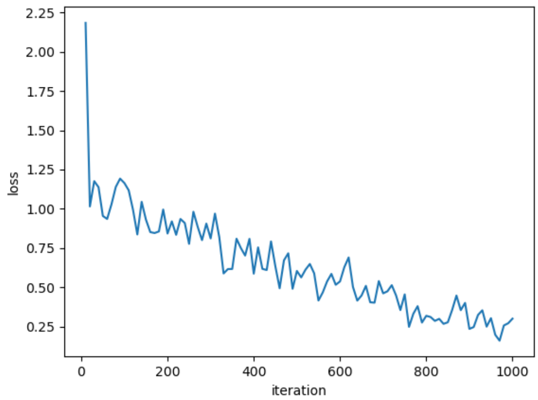

Figure 6.2: Contrastive loss after running for 100 epochs

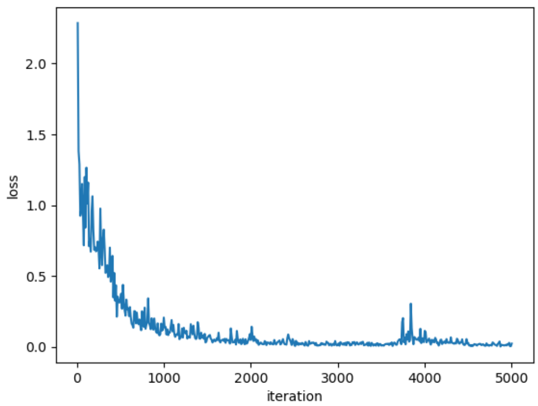

Figure 6.3: Contrastive loss after running for 500 epochs

We ran the model for 5000 epochs just to check whether the loss decreases further if the model sees the images more number of times. The results were not good because the loss was in the range 1.5 to 2.0. But when we trained for only 500 epochs, the loss at final stage hung around 0.05 to 0.02. This means that after 500 epochs, the model starts overfitting. 500 is the optimal value of number of epochs.

The same model when run on triplet loss function gives similar results, but the similarity scores are rather very high. A sample triplet is chosen and run through the network for 1000 epochs. The loss function gives low values of dissimilarity, but it doesn't seem to stabilize. We select contrastive loss as the loss function of the model because it gives better results.

Figure 6.4: A sample triplet batch

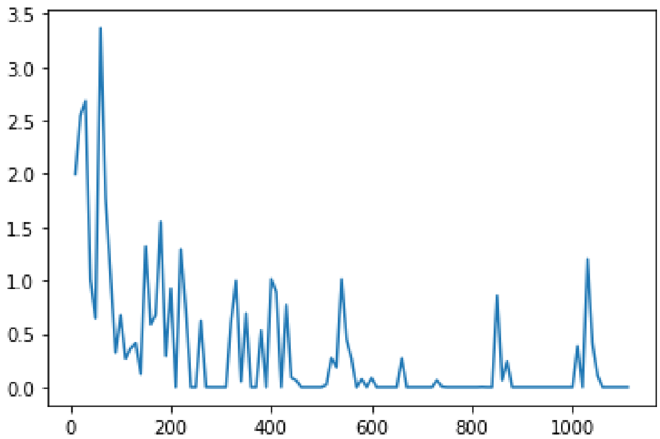

Figure 6.5: Triplet loss graph after running for 1000 epochs

6.3 Learning Rate

Learning rate of the model is tuned using cross validation. We found out that the loss is minimum at the learning rate of 0.0003. So, we selected it as the value of learning rate.

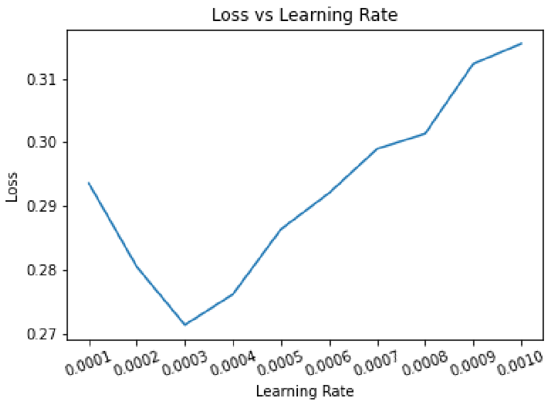

Figure 6.6: Contrastive loss for different learning rates

CHAPTER 7

Results

In Figure 7.1, we can see that the images of same person are having dissimilarity less than 2.0, whereas the images of different persons are having dissimilarity greater than 2.0. This is because we set the margin parameter in contrastive loss as 2.0.

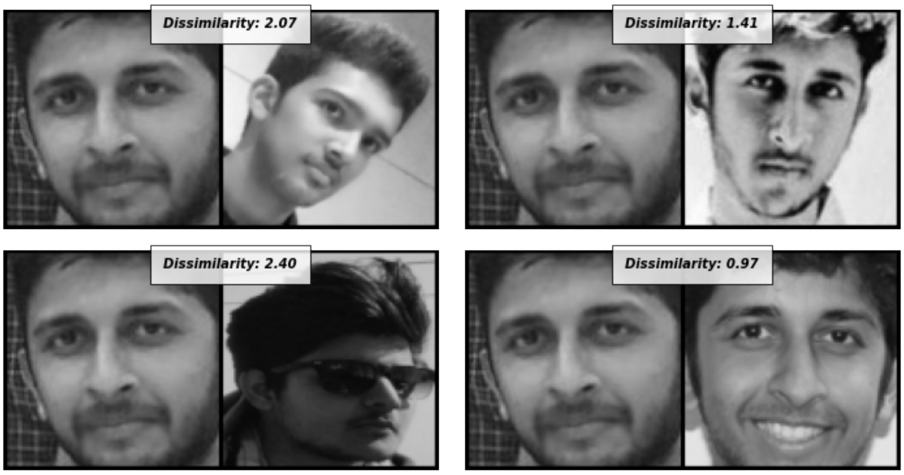

Figure 7.1: Test results using contrastive loss

Figure 7.2: Test results using triplet loss

Test results using triplet loss are given in Figure 7.2. We have selected contrastive loss as the loss function of the model because it gives better results.

In order to visualize how different classes are segregated during training, we used the encodings and extracted the top 2 PCA components. They are plotted on a scatter plot. This plot considers each component as x and y point in a plot. We hope that images with similar encodings will have closer PCA components. The visualization is done for 10 training epochs, and the results are shown in Figure 7.3. From the figure, it is clear that each class is getting grouped into the same cluster. This is an indication that the model is performing good. The figures are for epoch 2 to epoch 7 respectively.

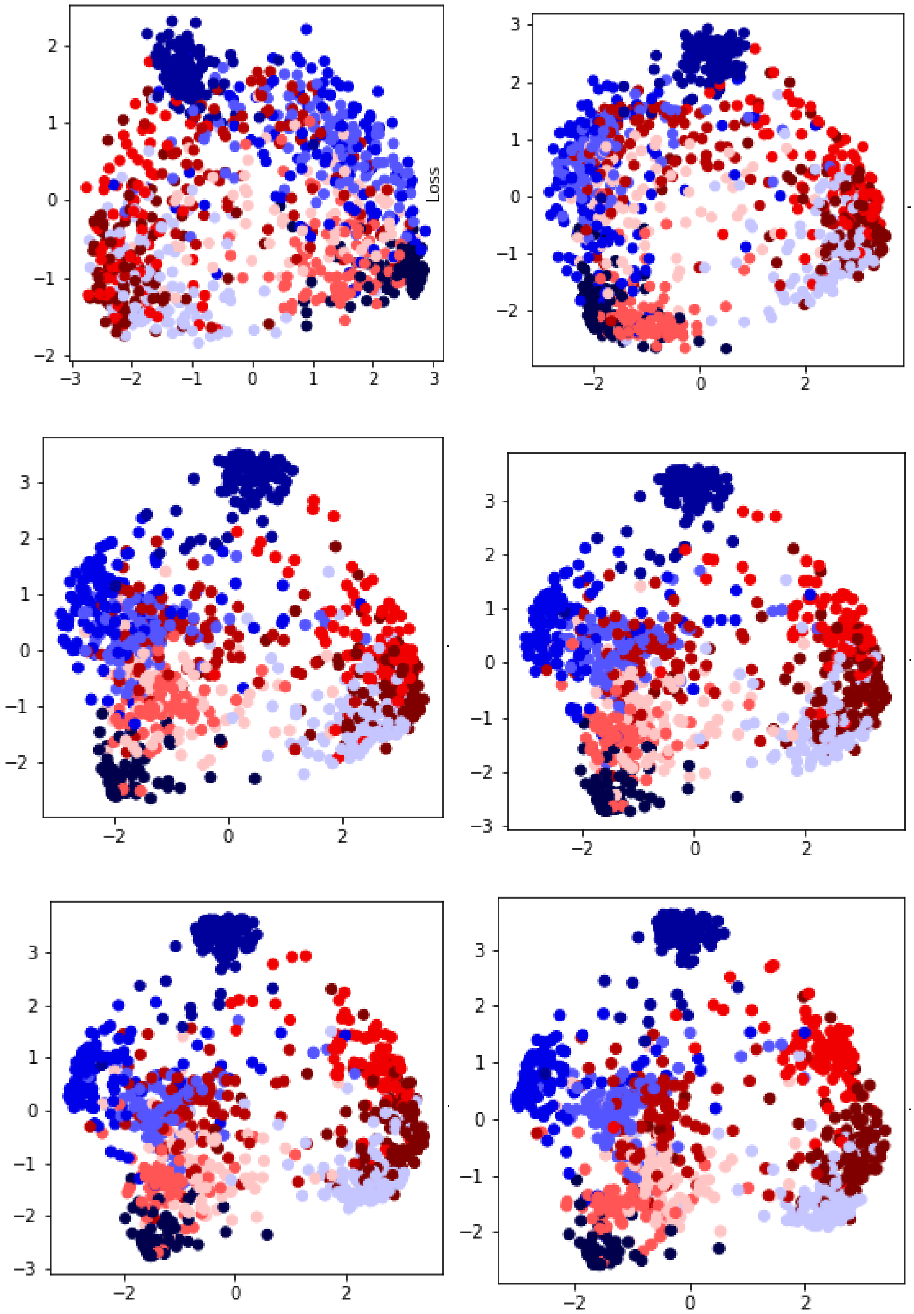

Figure 7.3: Test Results for epoch 2 to 7

7.1 Experimental Results on LFW Dataset

Labeled Faces in the Wild (LFW) is a database provided by University of Massachusetts. It contains 13000 photographs of faces. Each face has a label (name) associated with it. These faces were detected by using Viola-Jones algorithm. This dataset is used for evaluating performance of face recognition algorithms. The task is to find the name of the person in a given picture. Figure 7.4 shows how our model performed on this dataset. By iteration number 100, the loss becomes as low as 0.2. This is considered as a very good performance on LFW dataset.

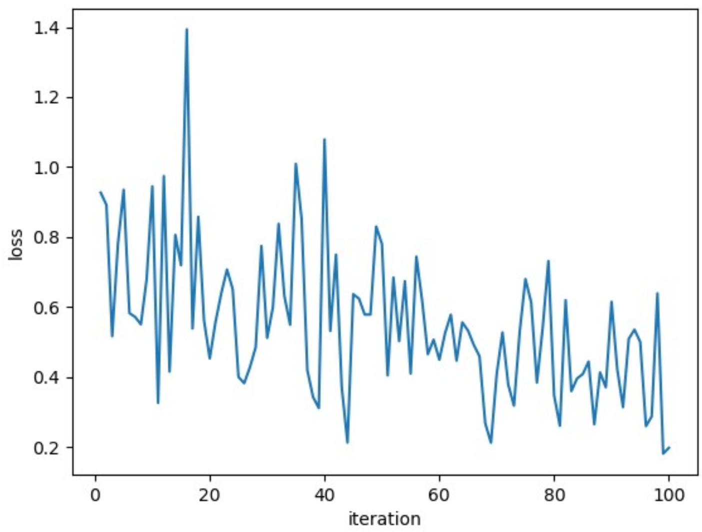

Figure 7.4: Performance on LFW dataset

7.2 Comparative Study with Existing Methods

Although we have successfully created a model inspired from FaceNet, we are slightly away from its accuracy (99.63% on LFW). This is because FaceNet contains a lot of parameters (approx 140 million) and was trained on multiple TPUs for several weeks. It performed about 1.6 billion floating point operations per second (commonly known as FLOPS). We did not have that infrastructure. Also, the size of trained model of FaceNet is in gigabytes(GB). Our aim was to create a model that isn't complex and produces small model size. In this case, our contrastive model worked inline with our expectations. The trained model size after 500 epochs is little less that 4 megabytes(MB) in size. This can be downloaded on any computer. One can always increase the accuracy of the model by incoporating state-of-the-art pretrained models like alexnet [7] and vgg19 [8] which have already proven helpful in many such cases. But the only problem here is the size of the trained model. Our focus was to create a simple model of small size and we have succeeded in our aim.

CHAPTER 8

Conclusion

In this project, an automatic attendance system has been implemented using Haar cascade classifier and one shot learning. We have developed a system that not only saves a lot of time of the lecturer but also reduces the amount of manual labour. The contributions made by our project are as follows : The proposed system eliminates the possibility of false attendance. It does not require any physical interaction with users. It reduces the wastage of time and manual labour significantly when compared to currently existing systems. Also, it is environment friendly as it eliminates the use of paper for attendance.

The proposed system turns out to be exceptionally valuable and will help in overcoming the limitations of traditional attendance systems if implemented on a large scale.

8.1 Future Scope

While introducing unseen images for testing purpose, the model was performing very good but there were some rare false predictions. These false predictions can be fixed by making further improvements in the project. We hope to increase the accuracy in future while keeping the size of trained model small. A python package can be created for this model which can be installed from pip package repository and anyone can perform face recognition using it. Also, one extra module for calculating attendance percentage of students can be included. It will reduce the workload of teacher to a bare minimum.

REFERENCES

[1] 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)*, Jun 2015.

[2] A. Jha, "Class room attendance system using facial recognition system," *The International Journal of Mathematics, Science, Technology and Management*, vol. 2, no. 3, pp. 4–7, 2007.

[3] V. Shehu and A. Dika, "Using real time computer vision algorithms in automatic attendance management systems," in *Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces*, pp. 397–402, IEEE, 2010.

[4] B. Surekha, K. J. Nazare, S. V. Raju, and N. Dey, "Attendance recording system using partial face recognition algorithm," in *Intelligent techniques in signal processing for multimedia security*, pp. 293–319, Springer, 2017.

[5] R. Hadsell, S. Chopra, and Y. Lecun, "Dimensionality reduction by learning an invariant mapping," pp. 1735 – 1742, 02 2006.

[6] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person reidentification," 03 2017.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, (Red Hook, NY, USA), p. 1097–1105, Curran Associates Inc., 2012.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.