ATTENDANCE SYSTEM USING FACIAL RECOGNITION

A PROJECT REPORT

Submitted by

Bharat Govind J TVE16EC019

V R Sreeganesh TVE16EC050

Amarnath A S TVE16EC007

to

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

in

Electronics and Communication Engineering

Department of Electronics and Communication

College of Engineering Trivandrum

Thiruvananthapuram

Kerala, India

July, 2020

Declaration

We undersigned hereby declare that the project report "ATTENDANCE SYSTEM USING FACIAL RECOGNITION", submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Prof. Neetha George. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place :

Date :

Bharat Govind J

V R Sreeganesh

Amarnath A S

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

COLLEGE OF ENGINEERING, TRIVANDRUM

CERTIFICATE

This is to certify that this report entitled **"ATTENDANCE SYSTEM USING FACIAL RECOGNITION"** submitted by **Bharat Govind J, V R Sreeganesh and Amarnath A S** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electronics and Communication Engineering is a bonafide record of the project work carried out by them under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Neetha George
Assistant Professor,
Department of Electronics and Communication,
College of Engineering Trivandrum.

Prof. Gijy P G
Assistant Professor & Project Coordinator,
Department of Electronics and Communication,
College of Engineering Trivandrum.

Dr. Santosh Kumar S, Ph. D.
Professor and Head,
Department of Electronics and Communication,
College of Engineering Trivandrum.

Acknowledgement

We take this opportunity to express our deep sense of gratitude and sincere thanks to all who helped us to complete the work successfully. Our first and foremost thanks go to God Almighty for giving us the strength, knowledge, ability and opportunity to undertake this project and complete it.

We wish to express our sincere thanks to Dr. Jiji C. V., Principal, College of Engineering, Trivandrum for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to Dr. Santhosh Kumar S (Head of the Department of ECE), for his support and co-operation. We wish to express our sincere gratitude towards all the teaching and non teaching staff members of our Department.

We wish to express our sincere thanks to our staff advisor, Prof. Joshua Thomas, for providing us with the support necessary for completing our project work.

We express our profound gratitude to our Assistant Staff advisor and Project Coordinator Prof. Gijy P G, who has provided us with her support and co-operation from the beginning to the end of this project work.

We are extremely indebted to our guide, Assistant Professor Neetha George, for providing us with priceless advice and guidance throughout the duration of project work.

Finally, we thank our parents, all our friends, near and dear ones who directly and indirectly contributed to the successful completion of this work.

Amarnath A S

Bharat Govind J

V R Sreeganesh

Abstract

Face recognition is one of the most promising applications the world of neural networks and machine learning has gifted the world. It is an incredibly versatile tool that has found its way to a large number of offices, schools and other institutions. Attendance registration is practiced in almost every workplace, schools and other similar institutions. Traditional methods, in general, are cumbersome and time consuming. We present a system which marks attendance while running silently in the background. The dataset was created by collecting videos of a number of students and sampling it frame by frame. Feature extraction was then performed on those images to obtain a dataset of features. This dataset of feature vectors were then used to train an artificial neural network to create a classifier. During deployment, the system captures an image from the class and isolates the faces present. The features are then extracted from the isolated features and those are then fed into the classifier to identify the student. The students who are present are marked as present in an excel sheet. Thus our system presents a smooth and zero effort method for marking attendance.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

There is a need for a convenient and efficient system that can perform the tedious task of marking students' attendance. Traditional roll call methods are exhausting, time consuming and susceptible to 'proxy calls'. Biometric scanning systems are expensive installations that lose functionality over time. Furthermore, students can 'scan and run' - scan their thumbprint and leave the premises. Considering all this, we thought that developing a novel system that could automate this process while maintaining accuracy would be a helpful tool for institutions, including our own.

While we are certain that other attempts have been made at tackling the same problem, we feel that our approach to the problem, which includes deep learning techniques, is a powerful, novel method. This same technology can be extended to other applications in related fields such as surveillance, security, biometric scanning, verification, etc.

1.2 Problem Statement

Mark attendance of students in a classroom using facial recognition methods, in all conditions including occluded and dark environments and independent of angle of capture.

1.3 Motivation

As students, we see attendance taking as an every day activity that requires our active participation. The method of attendance marking in our institution, and many other institutions, is roll call. This is a cumbersome physical process that is susceptible to cheating (proxy calling). Other methods include biometric scanners to mark the physical attendance of a person. This is widely used not just in many institutions but also at workplaces. Another method that is in use is a web check-in system. These methods all have their own pros and cons but inconvenience is a common feature. We propose an alternative, hassle-free method to mark attendance that requires only a student's physical presence and nothing more. Thus, our aim is to develop an efficient and convenient way to mark attendance.

1.4 Objectives

The primary objectives of the project are listed below:

1. Developing a facial recognition system
2. Optimising the system for non-ideal conditions
3. Develop a classifier to correctly detect faces in a populated environment
4. Accurately mark attendance

1.5 Scope

The scope of the project is to develop facial recognition system that can recognise faces despite a highly populated environment with chances of occlusion, classify the faces detected accurately and subsequently mark attendance of the student on this basis.

1.6 Outline of Report

- Chapter 1: Details the background of the problem along with motivation, objective and expected outcome.
- Chapter 2: Discusses the details of the existing systems along with its basic block diagram and working.
- Chapter 3: Details the block diagram and important parts of the proposed system.
- Chapter 4: Details the methodologies adopted while developing the system
- Chapter 5: Details the results and performance metrics of the system
- Chapter **??**: Discusses the possibilities of scaling this project, improving and developing into an integrated product
- Chapter 6: Conclusion

Chapter 2

Literature Review

2.1 Introduction

Over the years, deep learning and machine learning has found its way to a number of fields ranging from marketing to microbiology. One of the fields where machine learning has made itself indispensable is in the field of facial recognition.

Just like any other application out there, there are a variety of approaches to this goal. One such approach to face recognition is through feature extraction. This approach demands that features be learned first. This step is then followed by classification according to the features extracted from the image.

2.2 Related Works

Feature extraction means extracting a feature vector from an image. This feature vector facilitates classification of a particular class from others. The more complex or subtle the difference between the classes, the more number of features that needs to be learnt.

Autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner. The authors Lee et al.(2018) [10] used an autoencoder in order to enhance their two dimensional radar data. It was understood that autoencoders can be used to learn features in an unsupervised manner which we will be exploring in our work.

The image obtained and used for training the networks plays an important role regarding the performance of the networks. Sourcing image data thus is a stage that is of utmost importance. In the paper Illumination Invariant Face Recognition Under Near-Infrared Images [11] , authors Li et al.(2007) present a method to obtain images. This method enhances the classification and identification of images especially in non-ideal lighting conditions and their methods were incorporated in our system.

In the paper, Deep Residual Learning for Image Recognition [2], the authors He et al.(2015), presents an encoding network. This encoding network is based on the Res-Net 34 which takes the advantage of residual connections. In order to learn features used to identify faces, we need deeper networks. Deeper networks, intrinsically, face the problem of vanishing or exploding gradient descent. This problem is overcome by the use of residual connections. Residual connections add the output from a layer few connections before. This allows for identity mapping and overcoming the problem of vanishing or exploding gradient descent.

In the paper, Deep Metric Learning Using Triplet Network [1], the authors, Hoffer and Ailon (2015), presents a promising network that learns features that are necessary for classification. There's an implicit assumption that, when building a classifier, the network learns the features necessary for uniquely identifying members of a class. However, this network is designed so that it explicitly learns the features for uniquely identifying classes.

Having got an idea for how to extract features via encoding, the next step was to decide an approach for the classification problem. The paper [5] discusses the use of a Random Forest Classifier for large scale image classification. This application is similar to our problem of attendance marking, which includes classifying a large number of faces. The paper explores Random Forests and Support Vector Machines (SVMs) for classification. It explains a concept of 'incremental learning' and combining SVM and Random Forests for classification problems. The paper goes on to compare different methods and decision trees but the main takeaway from it is how we can use Random Forests for image classification.

The random forest method mentioned above is a machine learning method. We thought that it may be prudent to invest some time in a deep learning approach to this problem. The paper [6] discusses the use of a neural network for robust face detection and recognition. This was very helpful in understanding how we could adapt a custom built neural network to our image recognition problem. Following this, independent reading on classification using neural networks was done, additionally.

Chapter 3

Proposed System

3.1 Functional block diagram

The functional block diagram of the proposed system is shown in figure.

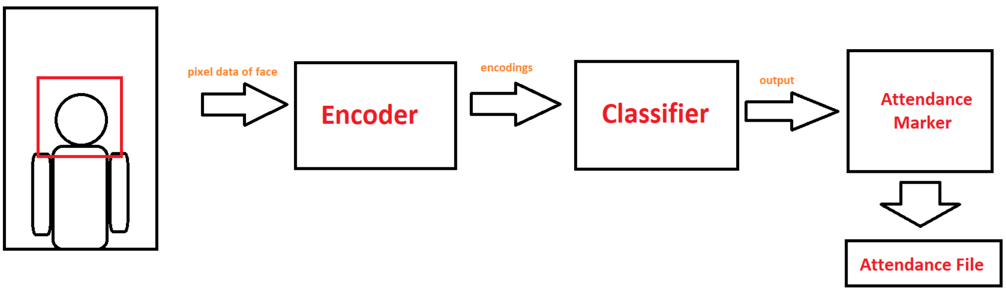


Figure 3.1: Block diagram of proposed system

To mark attendance, we take the image containing the student. The face is identified and a bounding box is drawn over it. We consider the area in the bounding box as the face.

Next, we feed the identified face to the encoder. The encoder converts the pixel data into a vector of encodings. For our project, we have decided to use a vector of 128 dimensions. The encodings are very small decimal values (float 32).

The encodings vector passes to the classifier. The classifier is a trained neural network that can identify the student from the encoding vector. The classifier is trained on a dataset comprising of the face encoding of students in a classroom.

The classifier outputs a value from 0 to n-1, where n is the number of students in the classroom. The value outputted is roll number or label number of a student. This means that that particular student is present in the image.

Using the classifier's output, we write to an excel file containing a record of students' attendance. A '1' is marked for the student that has been identified by the classifier. '1' indicates present and '0' indicates absent.

3.2 Working environment

Figure below shows the classroom setting with students and the camera.

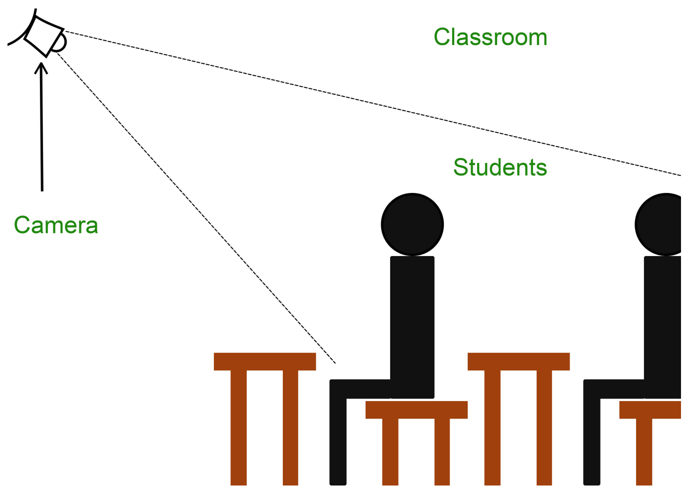


Figure 3.2: Classroom setting

The camera, placed at the top, at the very front of the classroom, faces the entire class. It captures an image, which contains all the students in the classroom. Then, faces are segregated using face detection in a cluttered environment. Each face is then fed into the system and based on the system output, attendance is marked.

Chapter 4

Architecture Description

This chapter includes the various methods adopted in developing each part of the system

4.1 Encoder

Encoder is any system or architecture that takes in an image and outputs a feature vector. The basic premise of an effective feature extraction network is that the features extracted from images of the same class lie closer to each other. Similarly the features extracted from images that belong to the separate classes are at a greater L2 distance from each other.

There are a number of different feature extraction networks. Two such approaches explored here are the auto-encoder networks and the triplet loss networks.

"Autoencoding" [10] is a data compression algorithm where the compression and decompression functions are data-specific, lossy and learned automatically. The auto-encoders explored here are fed an image and the network tries to reconstruct the image. The assumption is that while learning how to reconstruct the image, it would learn useful features necessary for classification.

The auto-encoders explored here was built using Convolutional Neural Networks. A number of architectures were explored and are as follows

4.1.1 Architecture 1

This autoencoder architecture consists of an encoder and decoder. The encoder has 20 convolution layers with the number of filters per layer ranging from 64 to 512. There similarly are 20 convolution layers in the decoder. The filter size in both networks are both 3 x 3. The architecture details has been given in the appendix. The dataset used to train this network is the LFW dataset [8].

4.1.2 Architecture 2

This autoencoder architecture consists of an encoder and decoder. The encoder has eighteen convolution layers with filter number ranging from 64 to 512. There are similarly eighteen convolution layers in the decoder with filter number ranging from 64 to 128. The filter size are all 3 x 3. The architecture details has been given in the appendix. The dataset used to train this network is the LFW dataset [8]. The input image is an image of size 128 x 128.

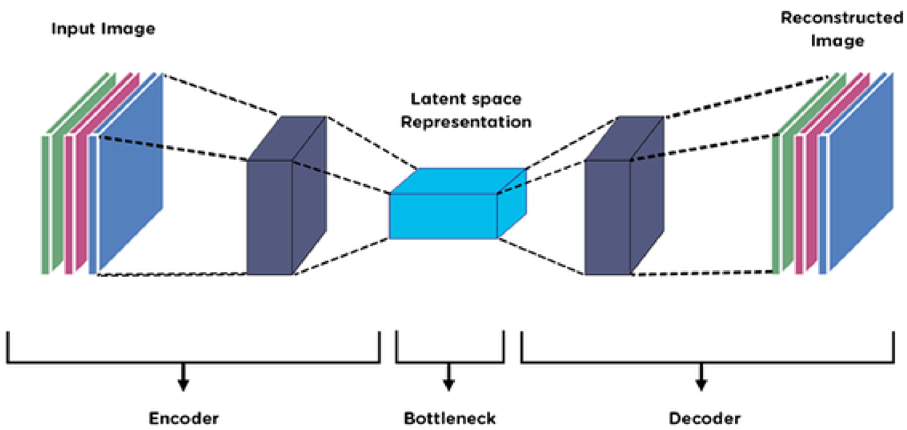


Figure 4.1: General Autoencoder Architecture [9]

4.1.3 Architecture 3

This architecture is that of a triplet network [1]. Here, the encoding network used has 14 convolution layers. The architecture details has been given in the appendix. The triplet network is fed three images. The images are labelled as "anchor", "positive" and "negative" sample.

During training, the "anchor" sample and the "positive" sample belongs to the same class while the "negative" sample belongs to another class. All three samples are fed to the encoding network. The encoding network which is a part of the triplet network extracts the encoding from the three images. The network is designed to reduce the L2 distance between the features of the "anchor" and "positive" sample and increase the L2 distance between the features of the "anchor" and "negative" sample.

The encoding network in the triplet network thus learns how to extract the relevant features. Once training is complete, only the encoder is used for feature extraction. Faces are extracted from an image and fed into this encoder in order to learn the features of that face.

The training loss after 50 epochs is 0.67 and the test loss is 0.7066. We can see from the errors here that the network did not converge.

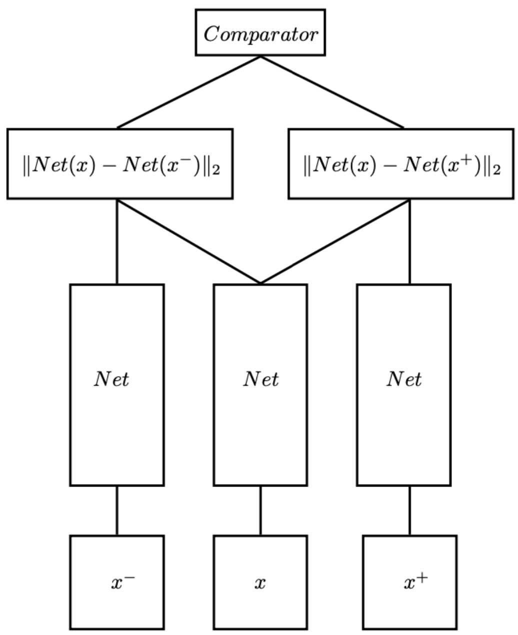


Figure 4.2: Triplet Architecture: Image source from paper [1]

4.1.4 Standard Trained Model

Though the auto-encoder architectures we've explored has shown good accuracy regarding reconstruction, the network hasn't learned the high frequency features. These features are the ones that are necessary for uniquely identifying a face. Thus these networks cannot be used for our application.

So, in order to extract the features from a face, we use the face recognition package designed by Adam Geitgey. This module uses a ResNet model trained by Davis King on a dataset of 3 million images. This network is 34 layers deep and has an accuracy of 99.38% when evaluated on the Labelled Faces in the Wild [8] dataset.

4.2 Classifier

4.2.1 Random Forest Classifier

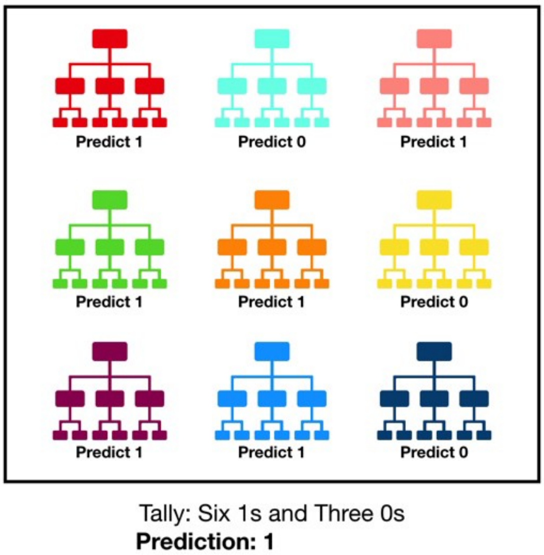


Figure 4.3: Conceptual representation of a Random Forest

Initially, a machine learning based classifier known as the Random Forest Classifier was used. It is a popular classifier model that is readily available in the open source machine learning library scikit-learn.

The way this classifier works is by pooling the results of multiple decision trees (hence the term forest). In the Fig 4.3, a representation of a random forest classifier is shown. Out of 9 trees, 6 predict 1 while 3 predict 0. Pooling the majority, the final decision is taken as 1.

The Random Forest Classifier can also be used for multi class classification. The function being readily available, all that was required was to feed in the data. The data we fed to this classifier were unprocessed arrays of encodings and corresponding name labels (String datatype). The data was split into training and test sets with a 60-40 split. Training was done and subsequently testing and predictions with test set. A 100% classification accuracy was observed. While this was outstanding, we realised that since this is a machine learning technique, it is possible that it performed so well because of the small size of data and labels. For scaling the system and making it more robust and precise, we would have to adopt deep learning techniques. The next section discusses the deep learning methods adopted that are now part of the final version of the system.

4.2.2 Dense Neural Network Classifier

A neural network was developed using the Keras library, with TensorFlow as a backend. The network was built using the Sequential Model class of Keras. The architecture of the network is shown in the figure below. A 128 dimension input layer is followed by 3 hidden layers of sizes 25,15 and 10 each with a 'ReLU' activation function, ending with an output layer of 9 units, with a 'softmax' activation function.

The dataset was preprocessed for this network. The encodings formed the X dataset. The labels, as Strings, were relabelled numerically from 0 to 8 for 9 classes. This formed the Y dataset. Both X and Y sets were split with a 60-40 ratio to form Training and Testing sets. No validation set was used owing to the low volume of data available.

The classifier output consisted of a vector of 9 values. These 9 values were processed to similar to reading a one-hot encoding scheme, wherein the maximum value in the vector was taken as 1 and all other values were rounded down to 0. After analysis of the vectors

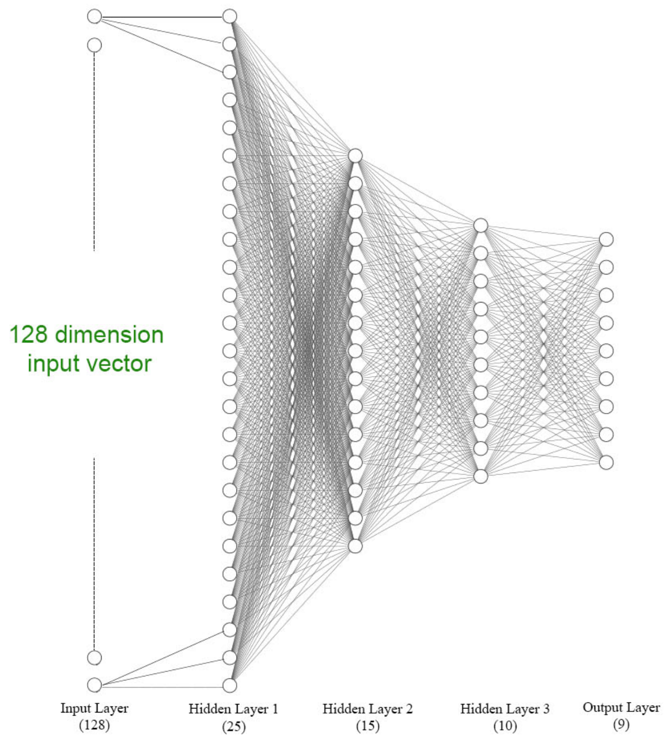


Figure 4.4: Architecture of the Classifier

generated, we found that the maximal values were always of the order 10E-1, while all other values were exceedingly small values.

These vectors were then converted back to the numeric labels (0 to 8) and compiled into a single vector. This vector was then analysed using a confusion matrix and a classification report. Analysis of the metrics revealed that the system was classifying with a very high accuracy, with very few misclassifications. The details of the results are discussed in Chapter 5.

4.3 Attendance Marking Scheme

The system was designed with the aim of marking attendance. Using excel files to mark attendance is a standard practice across institutions. Hence, the final stage our system is to create an excel file that has the attendance marked in a traditional and intuitive format as shown below.

Roll Number	Name	Date 1	Date 2	Date 3	Date 4	Date 5
1	Student 1	1	0	0	1	0
2	Student 2	0	1	1	0	0
3	Student 3	0	0	1	0	0
4	Student 4	0	0	0	1	0
5	Student 5	1	0	1	0	0
6	Student 6	0	0	1	0	0
7	Student 7	0	1	0	0	0
8	Student 8	1	0	0	0	1
9	Student 9	0	0	0	1	0

Figure 4.5: The Attendance Register

Chapter 5

Results And Discussion

5.1 Introduction

This chapter discusses the results and performance of the system.

5.2 Results and Performance of the Encoder

In order to obtain an encoder that was apt for our application, a number of architectures were explored and their performance was analysed.

5.2.1 Architecture 1

The first architecture which belongs to the class of autoencoders was shown to have a training error of 0.0063 and test error of 0.0072 after 50 epochs. The training and validation error graph is as shown.

5.2.2 Architecture 2

The second architecture which belongs to the class of autoencoders was shown to have a training error of 0.0430 and test error of 0.0435 after 40 epochs. The training and validation error graph is as shown.

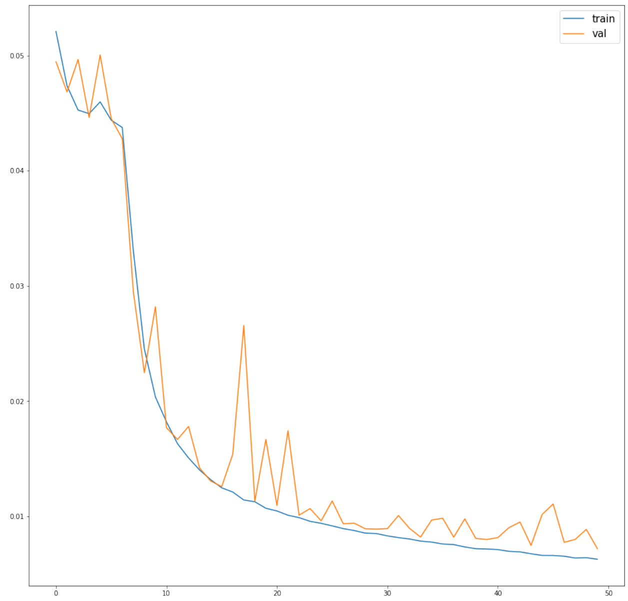


Figure 5.1: Training Loss for Architecture 1

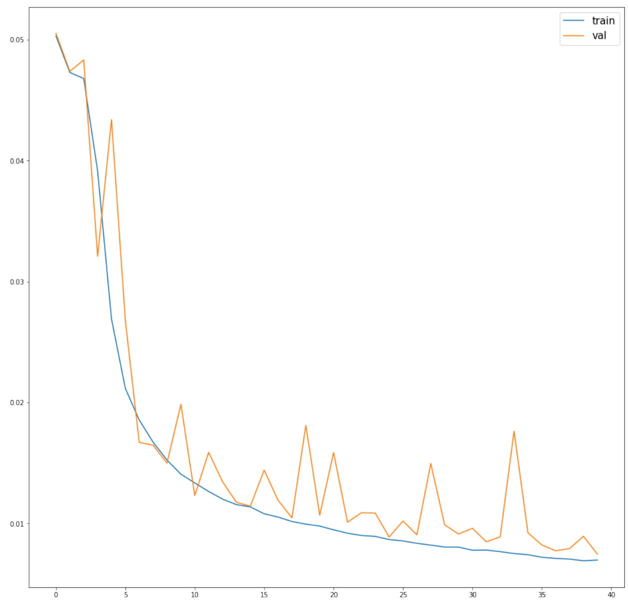


Figure 5.2: Training Loss for Architecture 2

5.2.3 Architecture 3

The third architecture which is used as the encoder in the triplet network was shown to have a training error of 0.67 and a test loss of 0.7066 after 50 epochs. The training and validation error graph is as shown

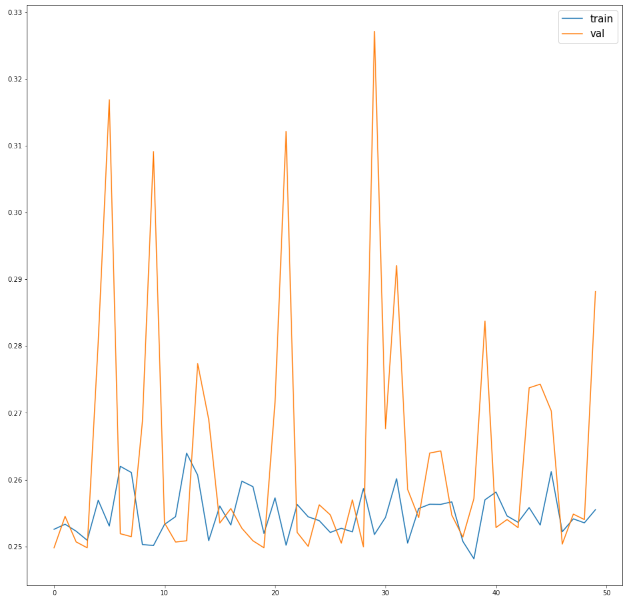


Figure 5.3: Training Loss for Architecture 3

5.2.4 Standard ResNet Model

This is a ResNet model trained by Davis King on a dataset of 3 million images. This network is 34 layers deep. This model is made available through the face recognition module made by Adam Geitgey. This network is shown to have an accuracy of 99.38 when evaluated on the Labeled Faces in the Wild(LFW) dataset.

```
The encoding of the image is
[array([-0.11670925,  0.07404245,  0.08041605, -0.0398895 , -0.04034629,
       -0.0961949 , -0.03420967, -0.01189145,  0.121525  , -0.03047629,
        0.10589866, -0.00219345, -0.14281636, -0.07307269,  0.02510006,
        0.04595257, -0.18820629, -0.16155736, -0.00877724, -0.08306167,
        0.0209115 , -0.00432562, -0.03502646,  0.06045636, -0.18777859,
       -0.3832601 , -0.09245859, -0.17193365,  0.02224641, -0.12003849,
       -0.02858767,  0.01851281, -0.17632049, -0.04651825,  0.00042418,
        0.13423745,  0.0157387 , -0.03410274,  0.139125  ,  0.08468549,
       -0.12058588, -0.05720334,  0.04084825,  0.3201589 ,  0.24292767,
        0.05419499,  0.00108757,  0.07051461,  0.12802166, -0.17842318,
        0.11388469,  0.10705628,  0.11796808,  0.09454119,  0.10326252,
       -0.11742569,  0.02819142,  0.02908383, -0.21932311,  0.05908785,
       -0.01073552, -0.06531106, -0.0230132 ,  0.03671226,  0.23158464,
        0.10329317, -0.0640961 , -0.07826809,  0.15867575, -0.23521082,
       -0.04998764, -0.01352999, -0.06622182, -0.0761108 , -0.24334866,
        0.03542724,  0.36103556,  0.13938177, -0.21542564,  0.02657217,
       -0.10328418, -0.09195115,  0.09811435,  0.05286713, -0.08876314,
        0.09437166, -0.11352044, -0.00659768,  0.19768818,  0.07244752,
       -0.0375101 ,  0.14903295, -0.02354675,  0.009328  ,  0.1313061 ,
       -0.04347503, -0.12345575, -0.04047235, -0.11988468, -0.10765569,
        0.03603081, -0.12745374, -0.06755923,  0.10656829, -0.21113551,
        0.11992569,  0.06043999, -0.03548717, -0.03960934,  0.07633881,
       -0.15341289, -0.11543931,  0.15694079, -0.22528374,  0.14097098,
        0.18354873, -0.01932883,  0.15194571,  0.02167189,  0.0315442 ,
       -0.02897046, -0.02578242, -0.07300415, -0.05260913, -0.00136568,
        0.02403293,  0.06142122,  0.03503641])]
```

Figure 5.4: Output of the Encoder

5.3 Results and performance of Classifier

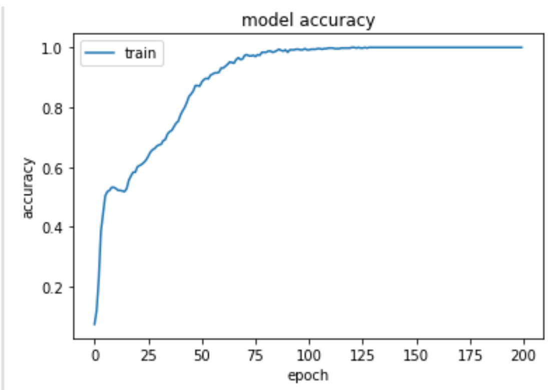
The classifier was trained with a batch size of 64 and 200 epochs. The training set had 494 examples.

The metrics monitored were loss function and sparse categorical accuracy.

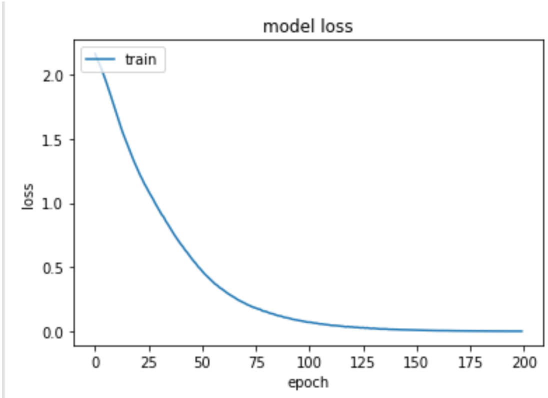
The final values of loss and accuracy were:

loss: 7.1696e-04 , sparse categorical accuracy: 1.0000

It is observed that the network converges around 125 epochs and has perfect training accuracy. The figures below show the convergence of the model accuracy and model loss.



(a) Model accuracy vs epochs



(b) Model loss vs epochs

Figure 5.5: Model convergence

The model was evaluated with the test set, which had 330 examples. The system showed the following values:

loss: 0.0555 , sparse categorical accuracy: 0.9848

Clearly, the system is performing very well.

To get a clearer idea on the performance of the system with respect to multi class classification, a prediction vector was generated containing all predicted labels. Then, a confusion matrix and classification report was generated.

Shown below is the confusion matrix for 9 classes (labelled 0 to 8).

```
[[20  0  1  0  1  0  2  0  0]
 [ 0 12  0  0  0  0  0  0  0]
 [ 0  0 55  0  0  0  0  0  0]
 [ 0  0  0 35  0  0  0  0  0]
 [ 0  0  0  0 41  0  0  0  0]
 [ 0  0  0  0  0 28  0  0  0]
 [ 0  0  0  0  0  1 97  0  0]
 [ 0  0  0  0  0  0  0 21  0]
 [ 0  0  0  0  0  0  0  0 16]]
```

Figure 5.6: Confusion Matrix

A near perfect diagonal matrix indicates that all classes are being classified perfectly, barring 1 or 2 misclassifications.

The next figure shows the classification report.

```
              precision    recall  f1-score   support

           0      1.000     0.833     0.909        24
           1      1.000     1.000     1.000        12
           2      0.982     1.000     0.991        55
           3      1.000     1.000     1.000        35
           4      0.976     1.000     0.988        41
           5      0.966     1.000     0.982        28
           6      0.980     0.990     0.985        98
           7      1.000     1.000     1.000        21
           8      1.000     1.000     1.000        16

    accuracy                          0.985       330
   macro avg      0.989     0.980     0.984       330
weighted avg      0.985     0.985     0.984       330
```

Figure 5.7: Classification Report

The report shows many metrics including precision, recall, f1 score and accuracy. The high values of precision and accuracy further reinforce that the system is working extremely well.

5.4 Final results

A set of images were tested in real-time to see if the system worked in cohesion to produce a classification and subsequently marked attendance.

The figures below show the use of a test image and the attendance being generated and written into an Excel file.

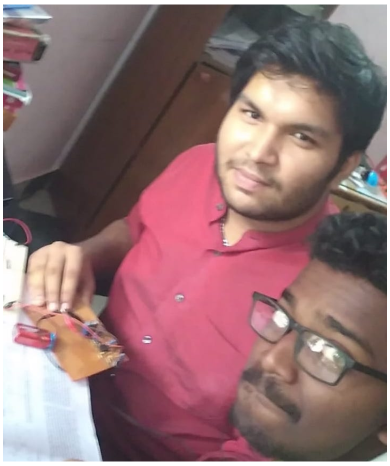


Figure 5.8: Test Image

Roll No	Name	Date 1
1	Adarsh	0
2	Athira	0
3	Midhun	1
4	Navami	0
5	Naveen	0
6	Oscar	0
7	Ramdas	1
8	Shiva	0
9	Sruthi	0

Figure 5.9: Test Image output

The attendance marked correspond to the names of the students in picture. Hence, system is working as expected with good performance.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

A convenient and hands-free method of attendance marking was developed using facial recognition and deep learning techniques. The system has good performance and can be scaled to accommodate higher volume of people in classrooms. The system finds applications in various other domains like surveillance, security, biometric scanning and verification of persons.

6.2 Future Work

6.2.1 Improving the system

The proposed system uses a 128 dimension encoding vector to perform the classification. This 128 dimension vector is a vector of features. However, this can be increased to a 256 dimension or 512 dimension vector. The advantage of increasing the feature vector dimension is finer features. Technology used to unlock mobile devices use similar high dimensional feature vectors to perform face recognition.

The problem with a high dimension vector is the space and computation involved. Considering the scale of the application, the computational and memory requirements can wildly vary. By default, considering a classroom environment, a high dimensional feature vector for a class of 60 students already becomes quite high. If it were to be used to monitor a room, say, a conference room, which can hold up to 150 people or more, the requirements are even higher.

However, this might be a necessary upgrade despite the costs involved. The camera that captures the image of the room is placed at a considerable distance away from a person's face. Extracting features in such a scenario can be difficult, because of the loss in quality. One way to solve this is to use a high dimensional feature vector.

Continuing along the same line, another improvement to the system comes in the hardware side. To supplement feature extraction, we can make use of a high quality camera, capable of capturing highly detailed photographs. This is a one-time cost that has permanent performance improvement on the system.

6.2.2 Scaling the system

Currently, the system we have can be described as being in experimental stages or very early stages. There is a lot of scope for scaling the system, for many applications.

This system was tested on a small set of 9 students, in very perfect conditions. The system was not tested under poor conditions as desired. The system can be scaled to accommodate a classroom of 60+ students. In such a situation, it is desirable to have an additional module in the system dedicated to segregation and detection of faces in a cluttered environment. In scaling the system, such a module becomes crucial to the success of the system as a whole.

While the current architecture of the neural network designed in this system is sufficient for most volumes of classification, it might be prudent to revise the architecture to better suit a different setting. Techniques like drop-out wherein a random set of neurons are dropped from the layers might help the network converge more efficiently. Adding layers and different activation functions might also show better results when scaling.

Bibliography

[1] "Deep Metric Learning Using Triplet Network" by Elad Hoffer, Nir Ailon

[2] "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun

[3] Scikit-learn Data Preprocessing Documentation
URL: `https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing`

[4] Scikit-learn Classifier Documentation
URL: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

[5] M. Ristin, M. Guillaumin, J. Gall and L. Van Gool, "Incremental Learning of Random Forests for Large-Scale Image Classification," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 3, pp. 490-503, 1 March 2016, doi: 10.1109/TPAMI.2015.2459678.

[6] C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pp. 1408-1423, Nov. 2004, doi: 10.1109/TPAMI.2004.97.

[7] Keras Sequential Model documentation
URL: `https://keras.io/guides/sequential_model/`

[8] Labeled Faces in the Wild @TechReportLFWTech, author = Gary B. Huang and Manu Ramesh and Tamara Berg and Erik Learned-Miller, title = Labeled Faces in

the Wild: A Database for Studying Face Recognition in Unconstrained Environments, institution = University of Massachusetts, Amherst, year = 2007, number = 07-49, month = October

[9] https://medium.com/@birla.deepak26/autoencoders-76bb49ae6a8f

[10] H. Lee, J. Kim, B. Kim and S. Kim, "Convolutional Autoencoder Based Feature Extraction in Radar Data Analysis," 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS), Toyama, Japan, 2018, pp. 81-84, doi: 10.1109/SCIS-ISIS.2018.00023.

[11] S. Z. Li, R. Chu, S. Liao and L. Zhang, "Illumination Invariant Face Recognition Using Near-Infrared Images," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 4, pp. 627-639, April 2007, doi: 10.1109/TPAMI.2007.1014.

Chapter 7

Appendix

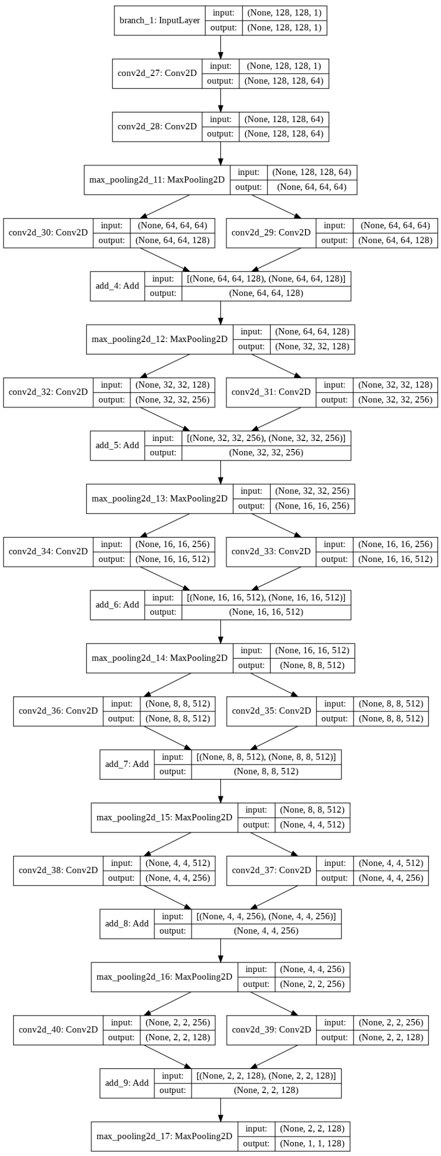


Figure 7.1: Triplet Encoder

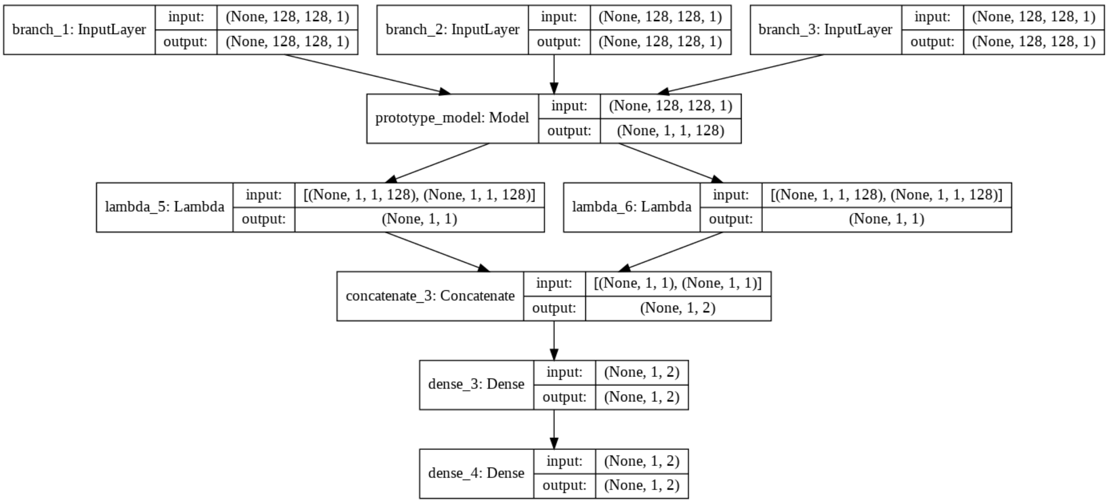
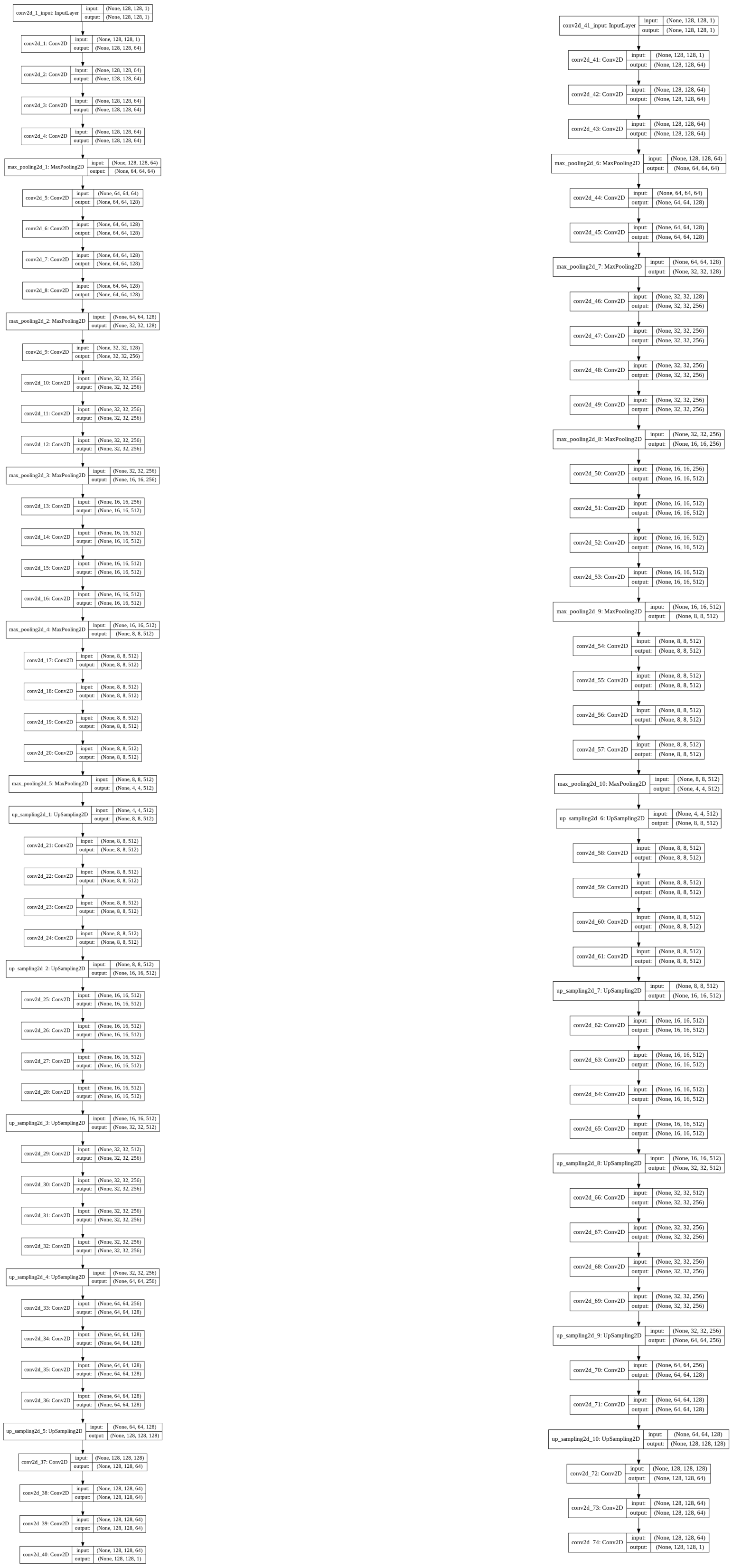


Figure 7.2: Triplet Architecture



(a) Architecture 1

(b) Architecture 2

Figure 7.3: Autoencoder Architectures