Study of Deep Neural Networks for Face Recognition

Raghunath Sai Puttagunta

Electrical Engineering

University of Missouri-Kansas City, MO-64110

rpyc8@mail.umkc.edu

Declaration- I Mr. Raghunath Sai Puttagunta, have written paper all by myself in my own words and all the literature and external knowledge used has been properly referenced in the Paper.

Abstract

In this paper we have used deep learning of neural networks for face recognition, we use data set provided by AT&T laboratories Cambridge. Initially we took only 6 Images from each subject for training and we used 2 layers for deep learning which were trained using auto encoders, and different parameters were tested and the parameters having the Highest ROC curves AUC was selected and then it was trained with soft max layer. Those 2 auto encoders and Soft max layer were stacked and a neural network was formed and that network was tested with the last 4 images of all subjects and ROC plots for that output of neural network were plotted and then conclusions about the accuracy of the system were made

Introduction

Face Recognition Algorithms are being in use for almost 50 years now. Many methods and algorithms have been used for implementing Face recognition systems. In this project we have implemented Facial recognition system using deep neural networks and with the data base provided by AT&T laboratories Cambridge. In this data base we have 40 subjects and each subject has 10 images. For some subjects the images were taken at different times so there was a variation in lighting, difference in facial expressions like open eyes or close eyes or smiling or not smiling and also different facial details like glasses or no glasses. All images of all subjects were taken against a dark background with subjects in an upright, frontal position. Each image is of the Size 92 x 112 pixels with 256 levels per pixel.

Face recognition is one of the important research areas in pattern recognition and machine learning. It's an incredibly challenging task to build an automated system which has more than or equal to human ability to recognize a person [2]. Face Recognition Systems can be used in biometrics to authenticate a person. From 1970 various algorithms like setting a few geometric parameters and trying perform pattern recognition using those parameters. The accuracy obtained by using that algorithm was around 45-70%, so different which isn't good enough. Principal Component is one of the major algorithm used in Face recognition Systems [2]. However in this paper we are discussing about implementing the Face recognition using deep learning of neural networks.

In this paper the discussion of results are done only using 2 layers for deep learning. Different number of layers can be used to implement this Face recognition system, but the parameters used should be carefully set for a higher accuracy in recognizing that person. This deep learning of neural network aspect is relatively new concept used for implementing Face recognition. The results obtained were highly accurate and were very close to ideal values.

In this paper in section 2 we discuss about the methods used for this algorithms and also the parameters that were used for deep learning. In section 3 we discuss about the results obtained and what does those result we obtained imply. We majorly discuss about the results of receiver operation characteristics ([1], page 388). In section 4 we discuss the conclusions we obtained from the results we obtained. In section 5 we give the references used. Section 6 contains appendix all the tables, figures and formulae which were used in this project.

Methods

In this project we were provided with a data base and that had 256 intensity levels per pixel. As neural networks only recognize 2 intensity levels we convert the 256 intensity levels to 2 levels. In the 40 subjects first 6 out of 10 images were used for training the auto encoder [3]. An auto encoder training neural network is unsupervised learning algorithm and uses back propagation ([1], page 149) for setting the target values to be equal to the input. ([3], page 12). For training an auto encoder we have 4 different parameters which can be changed to obtain an optimal result. The 4 different parameters are Hidden layer sizes, L2W regularizations, Sparsity Regularization and Sparsity Proportion. Auto encoder output will try to be almost same as the input. In this project each image is of 112 x 92 pixel and we are taking 6 Images from 40 subjects so the total size of the input layer is 10304 and if the hidden layer size is like 1000 so as the output of auto encoder is same as input the network will be forced to learn a compressed version of the input. Due to forcing of network for a compressed version of output we find some co-relation between inputs ([3], page 12).So different hidden layer sizes were used to obtain different configurations and receiver operations characteristics were plotted for different hidden layer sizes. L2W Regularization is another parameter which can affect the training of auto encoder. It's similar to weight decay it affects more for smaller weights while weight update after each iteration. ([1], page 246). Imposing a Sparsity constraint on hidden layers we can find interesting co-relation between inputs. Due to this we have 2 other parameters called sparsity regularization and sparsity proportion. Most of the neurons will be inactive in a network to change this we use sparsity proportion. The sparsity proportion is average number of activations over the sample. Due to this we add a term called sparsity regularization to cost function ([3], page 15).

We can use one or two auto encoders with different hidden layer sizes regularizations and different sparsity proportions. After training we encode the auto encoder to extract the features of the input. Later we take the Euclidean distance between images of same subjects and images of different subject. The Euclidean distance between same class are called intra class distances or intra class mean squared error and the Euclidean distance between different classes are considered to be inter class mean squared error. Intra class mse are called imposter and Inter class mse are called genuine. We try to maximize inter class variations and minimize intra class variations. We create targets for inter as +1 and intra as 0 and we plot the roc curve using the ezroc3 function. While giving he function to ezroc3 the concatenation of inter intra will be multiplied by -1 because we get smaller values of roc multiplying with -1 will flip it. Then we try to find the best configuration by varying those parameters and finding the best ROC curve with highest area under the curve. We use this configuration for soft max training of the neural network.

The features extracted from the auto encoder will be used to train the soft max layer then we use that network for finding the output for training and testing. For testing of the network we use the last 4 images of each subject. We create targets for testing and training separately. Using those targets we plot the ROC curve.

Results and Discussions

We try to get the best configuration of auto encoders by using different configurations of hidden layer sizes, L2W regularization, Sparsity Regularization, Sparsity Proportion and the best configuration is chosen on the basis of highest ROC area under the curve. Although theoretically ratios of mean squared errors can also be taken as one of the factor in deciding the best configuration but according to results obtained the ROC curves Area under the curve value looks more reliable than other parameters.

Initially we first start with different hidden layer sizes starting from 100 to 1000 using one auto encoder and extracting the features of the input. We can clearly observe that by plotting the plot weights of the auto encoder we can see better ghost images which means more features are being extracted. For all this values of hidden layers I got ROC curves AUC between the ranges of 0.93 to 0.946 (Refer Table 1). Logically we can see that higher size of hidden layer better the ROC but this might not be true because I got better ROC curve for 600 hidden layer size itself. It was around 0.9461 but the 1000 hidden layer sizes network was less than this. So by this only increasing hidden layer sizes would give better results isn’t true.

Also we can use 2 different auto encoders and stack them to get better results. In this process auto encoder 1 hidden layer sizes were varied from 100 to 1000 and auto encoder 2 hidden layer sizes were chosen from 100 to 500. All those configuration ROC’s were plotted and we see that the stacking of the auto encoders gave better results than one (Refer Table 2). The ROC curve area under curve were better than the using one encoder. So we can see that stacking both encoders gave us better results. All of these plots were plotted with optimal values of L2W regularization, Sparsity Proportion and Sparsity regularization. Also we can observe that if we plot weights of auto encoder 2 we observe the ghost images to be distorted. Also using the 3rd encoder didn’t improve the results much and considering the time taken for results only 2 encoder were used.

L2W regularization parameter is also one parameter in auto encoder varying the value can change the ROC curve. It’s generally low in the range of 0.004 it affects more for smaller weights than larger weights. ([1], page 209).So ROC’s were plotted for different values of weight starting from 0.004 to 0.016 with increase of weight of 0.004 in each step I observed that as the weights increased the ROC area under the curves were decreasing in general. (Refer Table 1)

Sparsity Regularization parameter was also tested for various values. We generally observe that as the value of the parameter increases the ROC area under the curves decrease. For Sparsity Regularization of 1 it was better than the Sparsity Regularization of 4. So we observe the trend that better ROC's for lower Sparsity regularization. (Refer Table 1)

Sparsity Proportion parameter as we increase this parameter the number of active neurons increases due to this when we increase this value we generally observe a better ROC. ROC's were plotted for three different value of Sparsity Proportion better values were obtained for higher Sparsity Proportion. So the optimum values I've used in L2W regularization, Sparsity Regularization and Sparsity Proportion were 0.004, 1 and 0.5 respectively. These were the optimum values used for the stacking of 2 auto encoders for obtaining the results explained in the above paragraphs. (Refer Table 1)

The best configuration obtained had a best roc of 0.95108 (Refer Table 2) this is for the training of auto encoder. Next we have to train the network using the soft max layer. We stack the auto encoder 1 auto encoder 2 and soft max. Next we use that deep net layer to find the values of output of neural network and plot ROC's we find that for training we get a perfect value of area under value which is 1 and the even decidability is high and has a value of 223444 and error rates has values in range of $10^{\wedge -16}$ (Refer Figure 11). So we can see that for train the results are highly accurate. Even if we try to check the output of neural network we observe that for that subject x we see that x^{th} row will be 1 for that image. All other value in that specific column will be closer to zeros. We got better values for soft max layer because it's a supervised training and whereas auto encoder is unsupervised learning that's why we get better results for soft max algorithm. While testing the best ROC area under the curve was 0.9425 we can fine tune the results by training deep net with training data and then testing it. The result which was improved was around 0.9894. While fine tuning the result was improved by only a fraction. Also the best 3 configurations were used to Plot ROC's and the average of each matrix is taken and that output is plotted for roc and we observe that the committee value was slightly lesser than the best configuration. (Refer Figures 9, 10) Also in this soft max layer we used 1 neural network and 40 outputs, this also can be done by taking 40 neural networks 40 outputs which means 1 output for each neural network (Refer Figure 5, 6, 7, 8). We observe that the result was almost 0.06 less than the result we got from the best configuration After observing the matrix we can see that some values are less than what they are expected it to be. This happens because all images were taken with different facial expressions of same person and some pictures were taken with or without glasses. So due to this network might make some errors in classifying images.

Conclusions

In this project first we have taken the data set provided by AT&T labs which had 256 levels per pixel and we converted it into 2 levels per pixel as neural networks only recognizes two levels. Next we try to train auto encoders with 6 Images from each subject using different parameters of auto encoders like different hidden layer sizes and different L2W and Sparsity regularizations and Sparsity proportions and found the optimum value of those parameters which happened to be 2 auto encoders of hidden layer sizes of 600 and 200 and L2w and Sparsity regularizations of 0.004 and 1 and Sparsity Proportion of 0.5. Later we used soft max layer to train the best configuration stack the auto encoder 1 auto encoder 2 and soft max layer together creating a deep network. Then we obtained training result and testing result. We obtained the training result ROC's area under the curve as 1 which is perfect curve which means the network perfectly classified the images. But while testing ROC area under curve was 0.984 which means the network made errors in classifying images. Although it has some errors 0.9894 accuracy is a very high value and it's almost close to the ideal values. Also one neural network 40 outputs is better than the results we obtained were for only this data set using different data sets might give lesser value of ROC area under curve. Also as the data size increases we might get bad results. Also using more layers while optimizing the values of parameters we might get better results.

References

1. J. C. Príncipe, et al., Neural and adaptive systems: fundamentals through simulations. New York: Wiley, 1999.
2. R Jafari, H.R. Arbiana , " A Survey of face recognition Techniques, Journal of Information Processing Systems Vol.5 No.2, June 2009
3. Sparse auto encoder by Andrew Ng
4. Simon O Haykin, Neural Networks and Learning Machine 3rd Edition, Prentice Hall, 2008.
5. C Gonzalez, R E Woods, S Liddins " Digital Image Processing using MATLAB

Appendices

Table 1- AUC's of 1 auto encoder with different Parameters.

Hidden Layer Size	L2W Regularization	Sparsity Regularization	Sparsity Proportion	Auc's	Ratio
100	0.004	1	0.25	0.94381	3.589
100	0.008	1	0.25	0.94256	3.850
100	0.012	1	0.25	0.93698	3.760
100	0.004	2	0.25	0.93943	4.37
100	0.004	4	0.25	0.91482	3.278
100	0.004	1	0.15	0.93352	4.789
100	0.004	1	0.50	0.94692	3.539
200	0.004	1	0.25	0.9442	4.459
400	0.004	1	0.25	0.94372	3.795
600	0.004	1	0.25	0.94670	3.910
1000	0.004	1	0.25	0.94231	4.194

Table 2-AUC’s using 2 auto encoders

Hidden Layer Size 1	Hidden Layer Size 2	AUC’s
200	100	0.94613
300	100	0.95097
300	200	0.94516
400	100	0.95324
400	200	0.94593
400	300	0.94363
500	100	0.94806
500	200	0.95099
500	400	0.94677
600	100	0.95028
600	200	0.95033
700	100	0.94973
700	200	0.95169
700	300	0.94443
800	100	0.95028
800	200	0.95056
1000	100	0.94733
1000	200	0.94999
1000	300	0.94664

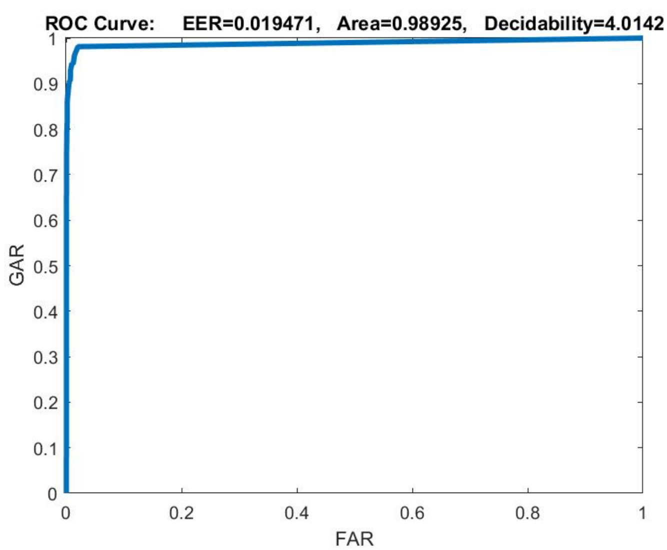

Figure 1 AUC Training (Soft max layer) (600,200)

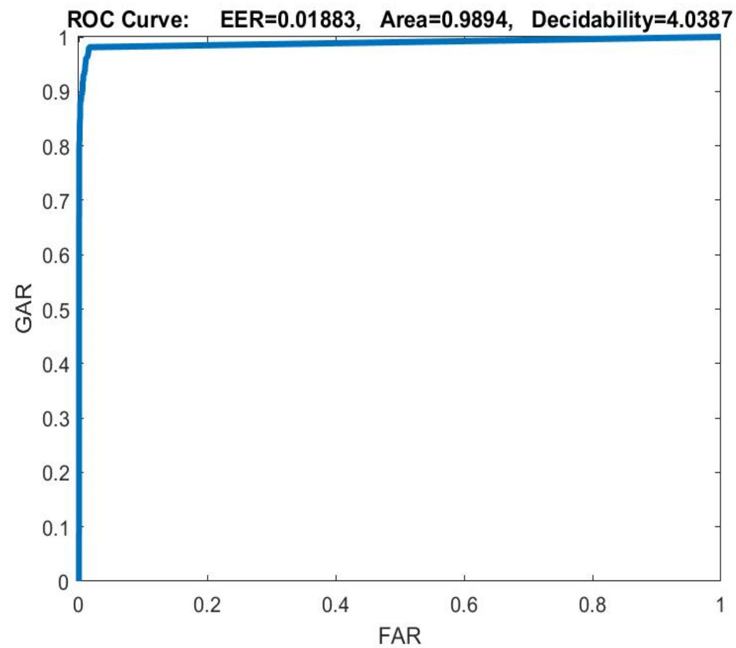

Figure 2 AUC's Training after Fine Tuning

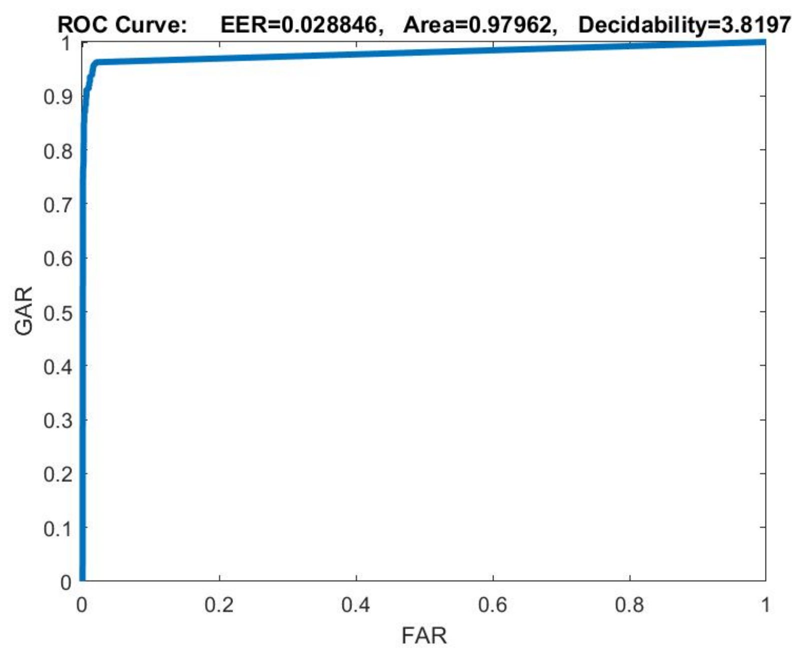

Figure 3- AUC Training (500,200)

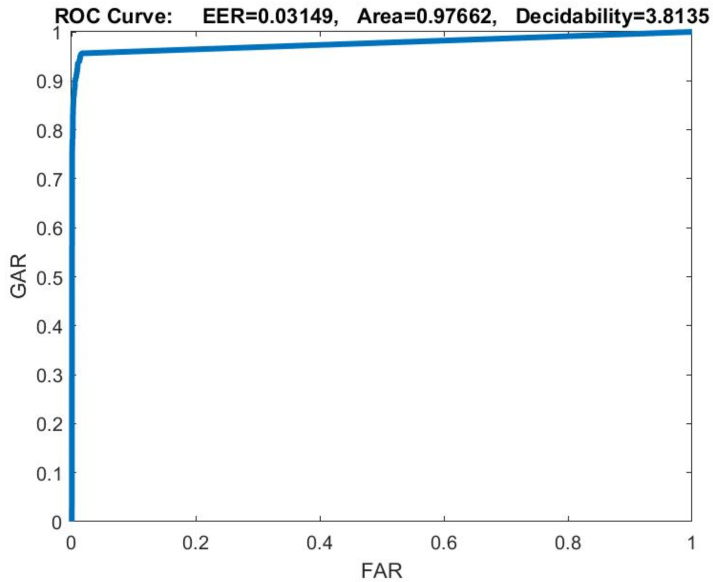

Figure 4 AUC Training after fine tuning (500,200)

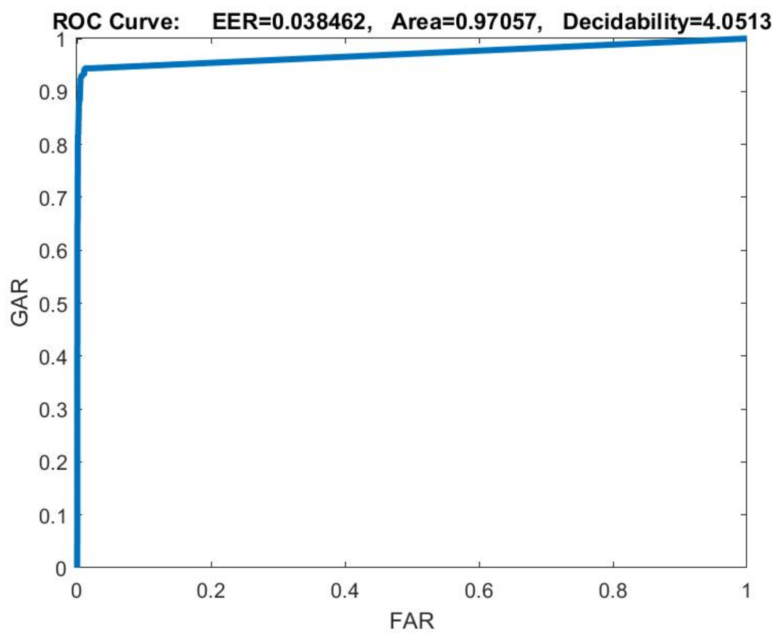

Figure 5 AUC Training (400,100)

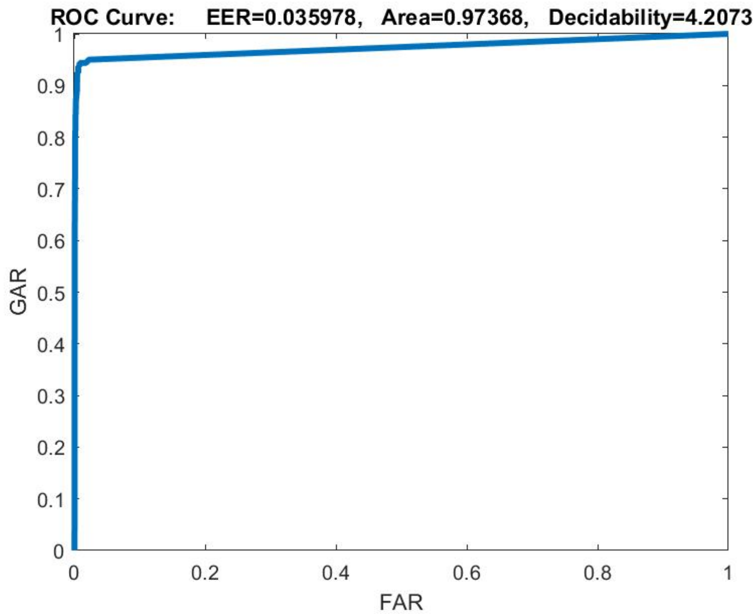

Figure 6- AUC Training after Fine Tuning (400,100)

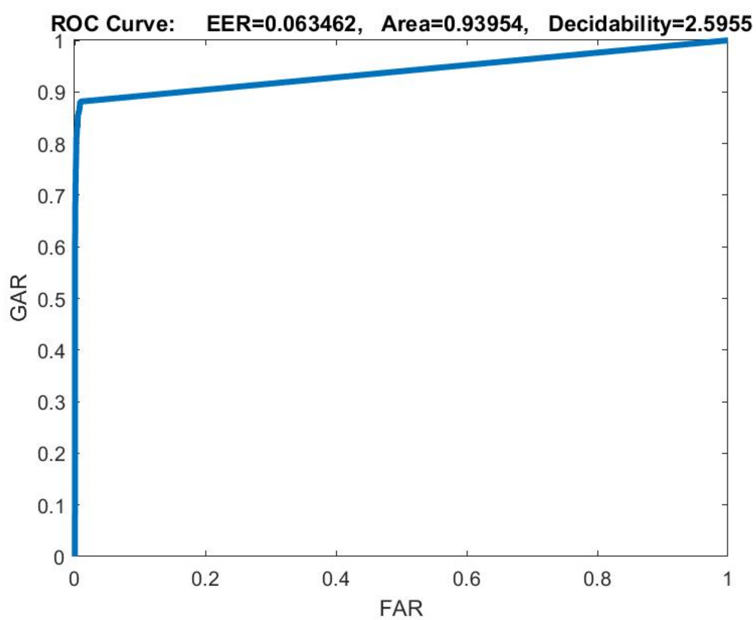

Figure 7-using 40 networks (400,100)

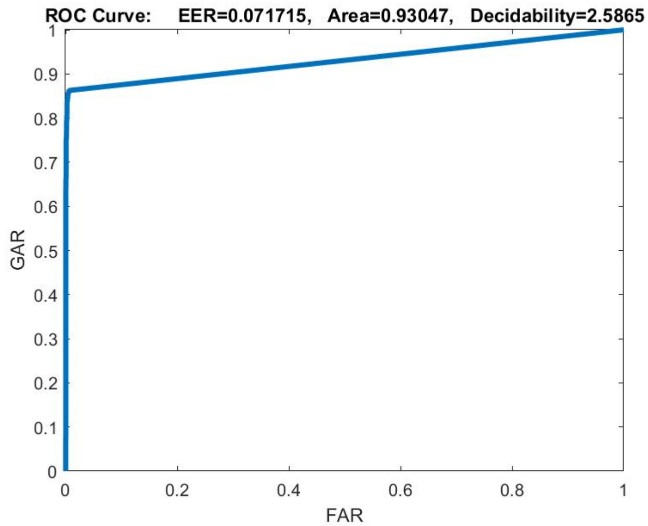

Figure 8 after fine tuning 40 networks (400,100)

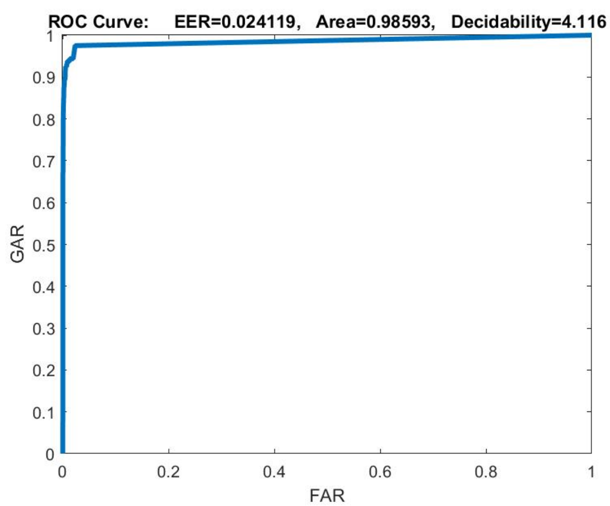

Figure 9 Committee after using 3 configurations

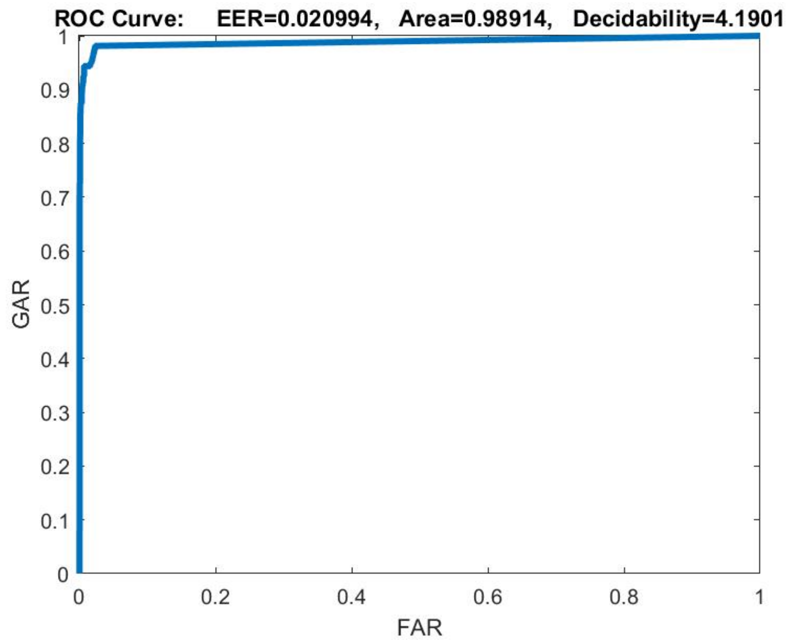

Figure 10 Committee after using 3 configurations Fine tuning

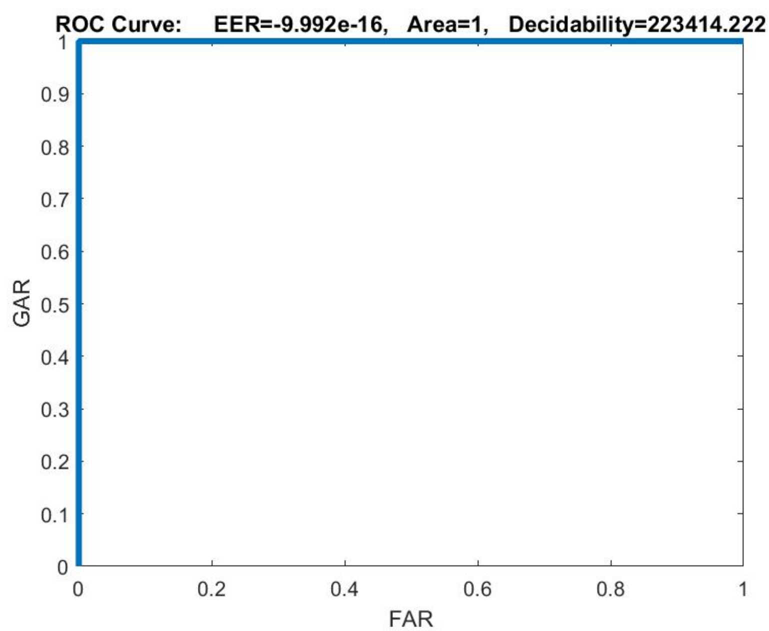

Figure 11- Train Soft max Layer