# Test Accuracy Improvement in Face Recognition Using Convolutional Neural Networks

M V Siva Prasad Chowdary, M Venkatanarayana

Abstract— Now-a-days face recognition plays a major role in identifying face of the specific person. There are different face recognition algorithms such as Eigenfaces algorithm, Local binary pattern histograms, Fisherfaces algorithm. All these algorithms face the problem of subject independence as well as translation, rotation, and scale invariance in the recognition of facial expression. In this study, the face recognition using neural network and convolutional neural network (CNN) techniques were utilized and implemented with the help of Python software 3.6.6. It is noticed that the test accuracy is improved against translation, rotation, and scale invariance in face recognition using CNN.

Index Terms—convolutional neural network(CNN), face recognition, python software, test accuracy.

### I. INTRODUCTION

Face recognition is advanced form of face detection. Instead of simply detecting faces in a given image or video it would be great that if we could specifically recognize the person. Face recognition in the field of computer vision refer to recognize the specific person's face in the image [3]. From the 2012 when image net challenge has been introduced a series of breakthroughs have been achieved. This is made more interesting with the inclusion of machine learning to find the solution of computer vision problems.

Face recognition is a process in which detecting a face in that particular image and then by using some algorithms to identify whose face it belongs to [2]. Now-a-days python programming language is a boon for implementing the face recognition. In python a library named OpenCV is a library that can be used actually it provides three methods for face recognition without using any machine learning technique. They are Eigenfaces algorithm, Local binary pattern histograms (LBPHs) and Fisherfaces algorithm [3]. All these three methods are used to recognize the face. These methods compare the required face that has to be recognized with some training set of known faces. If any unknown face is given that is not trained prior it uses the model trained on the training set to make the recognition. However the three methods use the training set differently.

Recently, motivated by the success of machine learning algorithms in computer vision. Initially neural network one of machine learning technique is performed on the dataset and later on the same data set convolutional neural network is performed. The CNN has made a series of breakthrough

# Revised Version Manuscript Received on 10 September, 2019.

M V Siva Prasad Chowdary, M.Tech Student KSRM College of Engineering, Kadapa, Andhra Pradesh, India.

(Email: mvspchowdary@gmail.com)

Dr. M Venkatanarayana, Dean & Professor KSRM College of Engineering, Kadapa, Andhra Pradesh, India.

(Email: mvnarayana@ksrmce.ac.in)

especially in the field of face recognition [7]. The primary aim of this analysis is to compare the test accuracies of the both neural network model and convolutional neural network.

#### II. TRADITIONAL METHOD

Eigenfaces Algorithm

The eigenfaces algorithm uses principal component analysis to construct a low dimensional representation of face images [1]. The principal component analysis (PCA) reduces the usage of large training data set there by reducing the computation expenses [3]. In general if we don't use PCA a lot of computation power is required which is in turn requires more time and a high end computer systems [2]. As Eigenfaces algorithm uses PCA it saves the data required, time and computational complexity of the systems [2]. The need for storing larger datasets of faces will not be required on using PCA [8].

The dimensionality reduction is achieved by PCA by choosing the linear projections that maximizes the scatter of all projected samples [9]. Those samples are chosen by PCA to represent a high dimensional training set into a lower dimensional subspace. The curse of dimensionality can be overcome by dimensionality reduction i.e. by reducing the dimensionality by selecting a subset of features rather than entire features [3]. It can be made possible by selecting a feature which is a combination of multiple features, in this way we can reduce the number of features required to train a model [12].

Consider a N-dimensional dataset(x) by finding a  $N \times K$  matrix (U):

$$y = U^T x$$
 where y has K dimensions and  $K < N$ .

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \xrightarrow{U^T} y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_k \end{bmatrix} (k < n)$$
(1)



# Test Accuracy Improvement in Face Recognition Using Convolutional Neural Networks

Eigenfaces maximize the scatter between all the face images in the training set. Here we use python programming for the assessment. Initially an empty folder will be created in the folder names datasets with the name of the specified person and then we run the gui.py python file then an GUI appears on the screen and in the enter the same name that is given prior. Later when the train button is clicked the camera app will be opened and it starts taking 100 pictures of the particular person.

Once after capturing 100 images of face an eigen\_train\_face.xml file will be generated automatically and all the features will be saved in that xml file and camera will be closed. After that when the recognize button on the GUI is pressed then the camera will be opened again and it starts recognizing the person. When a stranger tries to come in front of the camera then he would be recognized as unknown.

It looks very surprising with 100 images Eigenfaces algorithm could be able to recognize the face this is made possible only because Eigenfaces algorithm uses PCA. However there are some limitations in this algorithm. When the conditions such as light intensity or positioning of camera alters it could not recognize exactly. So in order to overcome these difficulties it would be better if neural network and convolutional neural network methods for face recognition.

#### III. PROPOSED METHODS

# A. Neural Network

Now a days machine learning algorithms has been made a drastic change in the computer vision problems. It has made a series of breakthroughs in the field of face recognition. Initially Neural network method will be performed on the dataset and later followed by convolutional neural network [3]. For any machine learning algorithm data is the most important. For any data to be used It should be pre-processed initially before using it.

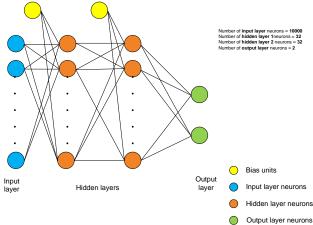


Fig. 1 Neural network architecture (with 1 input layer, 2 hidden layers and one output layer)

Here in the pre processing phase of the data two different data sets are taken one a boy and other a girl. Class labels have been assigned to this 0 for boy and 1 for girl. For boy we had taken 6,455 samples and for girl we had taken 6,225 samples of different face expressions. Then all the 12,680 samples are resized and converted the entire data into the CSV format we get 10,000 row values which indicate the pixel intensity values of a single image and then user convenience an additional class label of 0 and 1 are included

to the last column which makes it 10,001 rows likewise there are 12,680 samples i.e. that many number of columns in the generated data file.

In general a neural network model splits the given dataset into two portions. Here also one is a training set and the other is a testing set. 80% of the data fed to the neural network split as a training set data i.e. 10,142 samples and for test data 2,536 samples are taken randomly. In the neural network model the input is taken is fed at once into the input of the network. What happens is we fed the colour image(3 dimensional) to the system initially then while data pre-processing each colour image is converted to black and white i.e. a gray scale image(2 dimensional) and fed as an input to the neural network model. As the python programming language is used the index would start from 0 so the parameters would be is the train sample size and is the test sample size.

The samples used above are used for training purpose and when the class is to be predicted then

dense_1:Dense	Input:	(None,10000)
	Output:	(None,32)
dense_2:Dense	Input:	(None,32)
	Output:	(None,32)
dense_3:Dense	Input:	(None,32)
	Output:	(None,32)
dense_4:Dense	Input:	(None,32)
	Output:	(None,2)

Fig. 2 Neural network schematic block diagram.

When the total number of parameters that are to be learnt in the model are 3,20,032 parameters in input layer, 1,056 in first and second hidden layers. 66 parameters in the output layer. Altogether 3,22,210 trainable parameters are to be learnt including all the bias units also.

In the input layer (dense\_1) sigmoid activation function [0,1] is used. For first hidden layer (dense\_2) ReLU activation function is used and the same is used for the second hidden layer (dense\_3). At the output layer (dense\_4) softmax is used. For ReLU it forwards only the maximum value to the next layer. The final layer softmax would check the probability of both the classes and displays the class for which probability value is greater than 0.5 and displayed the class label 0-for boy and 1- for girl.



Here the total number of features required to train the model are more as the data set is less it can be performed easily if the dataset i.e. the size of the images increases or the number of samples increases the neural network model will require huge computational power and more computational time. So in order to reduce this are preferred.

#### B. Convolutional Neural Network

So At present CNN are showing tremendous results in the field of computer vision especially regarding face detection and face recognition which turned the attention of all the researchers towards CNN [6][5]. In the NN pixel intensity values of all the images are taken and concatenated into a single row and whenever it is required by the network an entire row is selected and given but where as in the CNN the image itself is taken directly by the model to process [11].

conv2d_1:Conv2D	input:	(None,100,100,1)
	output:	(None,98.98,8)
Max_pooling2d_1: MaxPooling2D	input:	(None,98,98,8)
	output:	(None,49,49,8)
,		
conv2d_2:Conv2D	input:	(None,49,49,8)
	output:	(None,47.47,16)
,		
Max_pooling2d_2: MaxPooling2D	input:	(None,47,47,16)
	output:	(None,23,23,16)
dropout_1:Dropout	input:	(None,23,23,16)
	output:	(None,23,23,16)
,		
flatten_1:Flatten	input:	(None,23,23,16)
	output:	(None,8464)
•		
dense_1:Dense	input:	(None,8464)
	output:	(None,32)
,		
dropout_2:Dropout	input:	(None,32)
	output:	(None, 32)
dense_2:Dense	input:	(None,32)
		(None,2)

Fig. 3 Convolutional neural network schematic model block diagram.

As the same data is used for both neural networks and CNN the data samples are also same but the number of features are different. Initially the features are extracted from the images and later by using the extracted features when they are passed through a NN model for predicting the class labels of the particular data [5]. So before feeding to the NN model for prediction some feature extraction procedure should be followed [10].

In CNN also the entire data set is divided into training (80%) and testing (20%). For training training samples are used represents the dimension of the image. Data

represents the testing data set shape these samples are given to feature extractor. After that these extracted features are taken as input parameters and fed as an input to the NN model

For the feature extraction convolution layer followed by max pooling layer likewise 2 pairs are taken followed by the dropout layer and then flatten is used. Later the extracted features are used as the input to the NN. In the first conv2D layer uses 8 layers of  $(3 \times 3)$  kernels and uses ReLU activation function for sub sampling Maxpooling layer of  $(2 \times 2)$  kernel is used.

The number of parameters to be learnt in conv2D\_1 are 80. Then in Maxpooling layer the size is reduced and no parameters would be there to learn. Then in con2D\_2 layer uses 16 layers of (3 x 3) kernels are used and the ReLU activation function is used. The total parameters obtained that are to be learnt are 1168. Then a max pooling layer is to be followed which uses (2 x 2) kernel for sub-sampling then a drop out of 0.25 is followed by flatten and the features of the particular image are extracted and they are fed as input to NN dense layer\_1 (hidden layer) and the parameters that are obtained to be learnt are 2,70,880 followed by another dropout layer 1 and finally given to the dense\_2 layer which uses SOFTMAX for predicting the class[5]. The total number of parameters that are to be learnt are 2,72,194. For the same data set for NN we need 3,22,210 parameters but at the same time CNN needs only 2,72,194 parameters

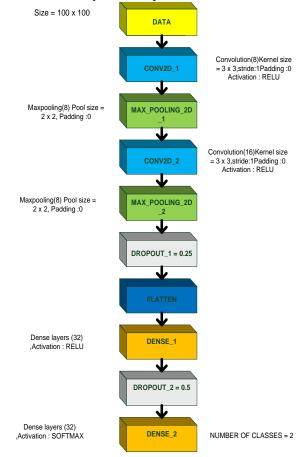


Fig. 4 Convolutional neural network architecture



#### IV. RESULTS AND ANALYSIS

The dataset that we used to compare the accuracies of both NN and CNN are created by slicing frames of the video. In order to get better results same dataset has been used. The total number of epochs used are 20 for both the NN program and CNN as well and the following results are obtained.

#### A. For NN Boy

```
Test loss: 0.1517301541679665
Test accuracy: 0.9428233438485805
Shape of First Layer (10000, 32)
Visualization of First Layer
[10.00838818 0.991611841]
Prediction
[10.00838818 0.99161184]
Thresholded output
[10 1]
Prediction Score:
[10.6528187 0.34718132]
Thresholded Score:
[1 0]
Predicted Digit:
0
```

Fig. 5 Result for the neural network boy class(0)

Figure 5 is the output result of the test accuracy of the boy which is class 0 with a test accuracy of 94.28% for 20 epochs

#### B. For NN Girl

```
Test loss: 0.09779668814423333
Test accuracy: 0.9416403785488959
Shape of First Layer (1000, 32)
Visualization of First Layer
[[1.8318960e-05 9.9998164e-01]]
Prediction
[1.8318960e-05 9.9998164e-01]
Thresholded output
[[0 1]
Prediction Score:
[1.8318960e-05 9.9998164e-01]
Thresholded Score:
[0 1]
Predicted Digit:
1
```

Fig. 6 Result for neural network girl class(1)

Figure 6 is the output result of the test accuracy of the boy which is class 1 with a test accuracy of 94.16% for 20 epochs

As there is a scope for improving accuracies the curse of dimensionality affects the speed and time of the computations. Generally neural networks works better with more data, as data increases the dimensions of the data also increases as the input dimensions increases the number of hidden layers increases. If the number of hidden layers increases which requires more computational power and more computational time. So in order to overcome the curse of dimensionality CNN is preferred.

### C. For CNN Boy

Fig. 7 Result for convolutional neural network boy

#### class(0)

Figure 7 is the output result of the test accuracy of the boy which is class 0 with a test accuracy of 95.26% for 20 epochs

#### D.For CNN Girl

```
Test loss: 0.08708073222953162
Test accuracy: 0.9542586750788643
Prediction Score:
[2.0023707e-08 1.0000000e+00]
Thresholded Score:
[0 1]
Predicted Digit:
1
```

Fig. 8 Result for convolutional neural network girl class(1)

Figure 8 is the output result of the test accuracy of the boy which is class 0 with a test accuracy of 94.28% for 20 epochs

Table I.

Test accuracies of neural network and convolutional neural network.

Classes	Test Accuracy	
	Neural	Convolutional
	Network	Neural Network
Boy(0)	94.28233	95.2681
Girl(1)	94.1640	95.4258

From the results, it is noticed that the accuracy of face recognition is increased using CNN.

## V. CONCLUSION

By using Eigenfaces algorithm which was used initially used for face recognition which uses 100 samples of a particular person to detect but if any variations happens in placement of camera or light intensity it will not detect the face. Actually it uses PCA that is the reason this algorithm uses only 100 face images to detect.

Machine learning algorithms have clearly shown the great improvement initially we used NN algorithm has used to detect the faces of the person, to some extent we get good test accuracy initially but due to the curse of dimensionality we cannot get the better accuracy because in order to increase accuracy we require more data which leads to increase in the computational power and computational time.

CNN uses feature extractor used to extract the features of the images of faces. Then the required numbers of features that are to be trained are reduced when compared to the NN and we get better test accuracy when compared to NN. From results, it is observed that CNN provides better results for computer vision problems especially for face recognition.

There is a scope to improve the test accuracy using CNN by increasing the size of data and computational power.

# VI. ACKNOWLEDGMENT

The authors are very thankful to the Center for Research and Innovation KSRMCE for providing facilities to complete this paper in supporting me to learn and implement ANN and



its variants using the facilities available at CRI in K.S.R.M College of Engineering.

# REFERENCES

- Mayank Agarwal, Himanshu Agarwal, Nikunj Jain and Manish Kumar, "Face Recognition using Principal Component Analysis, Eigenface and Neural Network", International Conference on Signal Acquisition and Processing, pp. 310-314, 2010.
- Adrian Rhesa Septian Siswanto, Anto Satriyo Nugroho and Maulahikmah Galinium, "Implementation of Face Recognition Algorithm for Biometrics Based Time Attendance System", Center for Information & Communication Technology, Indonesia, 2014.
- I. Paliy, A. Sachenko, V. Koval and Y. Kurylyak, "Approach to Face Recognition Using Neural Networks", IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 112-115,2005.
- Kewen Yan, Shaohui Huang, Yaoxian Song, Wei Liu and Neng Fan, "Face Recognition Based on Convolution Neural Network", Proceedings of the 36th Chinese Control Conference, Dalian, China, pp.4077-4081, July 26-28, 2017.
- N. Pattabhi Ramaiah, Earnest Paul Ijjina and C. Krishna Mohan, "Illumination Invariant Face Recognition Using Convolutional Neural Networks", Indian Institute of Technology Hyderabad Andhra Pradesh, India, 2015.
- Musakuzu Mutsugu. Katsuhiko Mori, Yusuke Mituri and Yuji Kntiedu, "Facial Expression Recognition Combined with Robust Face Detection in A Convolutional Neural Network", Canon Research Center, Japan, 2003.
- Hao Wu, Ke Zhang, and Guohui Tian, "Simultaneous Face Detection and Pose Estimation Using Convolutional Neural Network Cascade", IEEE Access, IEEE, 2018.
- Convolutional E. Bingham, H. Mannila. "Random projection in dimensionality reduction: applications to image and text data", Proceedings of the 7th International conference on Knowledge Discovery and Data Mining, San Francisco, USA, 2001, pp. 245-250.
- P. N. Belhumeur, 1. P. Hespanha, D. 1. Kriegman, "Eigenfaces vs. Fisherfaces:Recognition Using Class Specific Linear Projection", European Conference on Computer Vision. 1996.
- Jonathan Dowdall, Ioannis Pavlidis, and George Bebis, "Facedetection in the near-ir spectrum," Image and Vision Computing,vol. 21, no. 7, pp. 565-578, 2003.
- AR Syafeeza, M Khalil-Hani, SS Liew, and R Bakhteri, "Convolutional neural network for face recognition with pose and illumination variation.," International Journal of Engineering & Technology(0975-4024), vol. 6, no. 1, 2014.
- E. Bingham, H. Mannila. "Random projection in dimensionality reduction: applications to image and text data," Proceedings of the 7th International conference on Knowledge Discovery and Data Mining, San Francisco, USA, 2001, pp. 245-250.

