International Journal of Engineering Sciences Paradigms and Researches
(Vol. 39, Issue 01) and (Publishing Month: February 2017)

Human Face Recognition using CNN Method

Gati Krushna Nayak[1], Mohini Prasad Mishra[2], Pratyush Ranjan Mohapatra[3]

[1,3]Associate Professor, Department of Computer Science Engineering, Gandhi Institute For Technology (GIFT), Bhubaneswar

[2]Assistant Professor, Department of Computer Science Engineering, Gandhi Engineering College, Bhubaneswar

Publishing Date: February 27, 2017

Abstract—Face recognition is the process of assignment of correct label to the face under consideration. Process of face recognition comprises of extraction of features from underlying face and feeding the extracted features to classifier to identify the corresponding individual. The Accuracy of classifier is greatly affected by the nature of features extracted. Conventional face recognition system relies on manually engineered, handcrafted features. Convolutional neural network is a deep learning model which automatically extract features from the raw data in the process of end to end classification. The automated feature extraction property of convolutional neural network not only saves the effort in manually extracting features but also solve the dilemma of set of features to be used for classification. In this work, we present a face recognition system based on convolutional neural network. We create our own dataset to test the efficiency of the proposed system. The accuracy of around 96% is achieved on the test dataset consisting of around 1900 images with 10 different classes.

Keywords—Face recognition, Convolutional neural network, Feature extraction

I. Introduction

Face recognition is the process for identifying an individual on the basis of its facial features. It has numerous practical applications such as biometric attendance marking, surveillance systems, Face ID etc. Face recognition can be treated as a classification problem. A typical classification problem consists of two phases. In training phase, the discriminant features are extracted from training data. Then, a classifier is designed which operates on the features of the training data. The classifier is trained to minimize the discrepancy between the class predicted by the classifier and the actual class of the corresponding training data. In the test phase, features are obtained from the test object using the same mechanism used for training data. These features are then fed to trained classifier to know the class to which test object belongs to. To increase the accuracy of the whole system, the extracted features and the designed classifier should complement each other. In other words, features should be discriminant enough for the classifier to make correct decision and the classifier should be competent to separate the features belonging to different categories.

In conventional Face recognition systems, the facial feature extraction process can be classified into geometric featurebased process and appearance feature based process [1]. Examples of geometric feature-based process include active appearance model [2] and local global binary pattern (LGBP) [3]. Appearance based facial feature extraction techniques include Principal component analysis (PCA) [4], Linear discriminant analysis (LDA) [5], Local binary pattern (LBP) [6], Gabor wavelet transform [7], Fisher discriminant analysis (FDA) [8] and Speed-Up Robust Features (SURF) [9]. The commonly used classifiers for face recognition involve Knearest neighbor [10], artificial neural networks (ANN) [11], support vector machines (SVM) [12] and Hidden markov model [13].

In recent years, deep learning has tested enormous success in the field of computer vision. Specially in the field of classification, deep learning modules have shown upper hand compared to most of the conventional systems in terms of accuracy. Deep learning modules like ANN and Convolutional neural network (CNN) are found suitable for supervised classification (where one has the access to labels of training data) and for unsupervised classification (where one does not have the access to labels of training data) Autoencoder model step up.

ANN module consists of input layer, hidden layer and the output layer. Input layer consists of nodes equal in number to the samples of one input data example. Each node in the input layer represent the value of corresponding sample of a particular input data example. Hidden layer nodes are representative of features of the input data. There can be several hidden layers to learn hierarchical features. The output layer consists of nodes equal in number to the classes. The parameters of ANN model are trained to minimize the distance between the actual training labels and the estimated labels by ANN [14].

ANN models are efficient for data in which there is no spatial correlation. For the data like images, it is necessary to maintain the spatial correlation between pixels while extracting features. CNN [15] model extracts the features from the images using convolution operation while maintaining the spatial correlation. In the end to end training process, CNN model can extract the discriminant features from the raw input data. The autonomous feature extraction capability of CNN eliminates the need of hand-crafted feature extraction. CNN, on account of its ability of autonomous feature extraction not only saves the effort in manually extracting the handcrafted features but also solve the dilemma of which set of discriminant features to use.

Counting on the automated discriminant feature extraction capability of CNN, we implement a CNN model for the task of face recognition. We create our own dataset to train the CNN model.

A typical CNN model mainly consists of convolutional layer, Pooling layer and the fully connected layer. In convolutional layer, convolutional filters are convolved with previous layer to form the feature maps. Pooling layer is used after the convolutional layer to reduce the dimensionality of feature maps. There are various mechanisms of pooling such as max and mean pooling. For instance, in max pooling only the neuron with highest value in specified pooling size is retained. The fully connected layer part of CNN resembles to ANN. The first layer of fully connected part is formed by reshaping the feature maps of the previous layer into a vector. The first layer of fully connected part is followed by several fully connected hidden layers. The ultimate layer i.e output layer of CNN consists of neurons equal in number to that of classes under consideration. The nodes in the output layer of CNN are indicative of the classes estimated by CNN. A loss function, which define the discrepancy between the classes estimated by CNN and the true classes is defined. While training, the parameters of CNN are tuned towards minimizing this loss. The common examples of loss functions are mean squared error, softmax cross entropy and sigmoid cross entropy loss.

II. Dataset and Architecture

A. Dataset

A proper labelled dataset is one of the integral parts of neural network training. A dataset must be large enough to train the network to make it scale invariant, also it should be able to identify faces in different lighting conditions and angled at various angles.

We created the dataset using the webcam by taking about 200 images per individual, in different lighting conditions with different poses and various facial expressions. Total of 10 individuals took part in the process. All the images were scaled to a 128×128 pixel size. To expand the dataset, we did image augmentation by flipping the images along their vertical axis. Rotating the images randomly about their center, clockwise and anticlockwise up to 40º. To make the model robust to noisy images, different noises were added to the training dataset. Gaussian noise with a variance of 0.01 was added. Poisson and Speckle noise with zero mean and variance of 0.02 was added .

In total around 800 images of each individual were obtained by image augmentation methods, leading to a total of 7767 images consisting of 10 different classes. 60% of the complete dataset was used to train the CNN, 20% was used for validation and the remaining 20% was used for testing purpose.

Figure 1 depicts a sample dataset.

B. Architecture of CNN

The proposed architecture of CNN consists of three blocks of convolutional and pooling layer. Each block consists of two convolutional layers followed by a pooling layer. In first block, the convolutional filter size and the number of feature maps were set to 5×5 and 32 respectively, for both the convolutional layer. In second and third block, for both the convolutional layers, the convolutional filter size and the number of feature maps were set to 3×3 and 64 respectively. Max pooling mechanism was used for pooling layer of all the three blocks with the pooling size of 2×2. The leaky relu activation function was used for all convolutional layers. The output layer of third block was flattened to create a column of neurons. The flattened layer was followed by a fully connected layer consisting of 1024 neurons. The network ends with the output layer consisting of 10 neurons (10 classes). Figure 2 shows the proposed CNN architecture.

C. Training

The proposed CNN model was trained towards minimizing the softmax cross entropy loss. We used batch wise training process with the batch size of 10. The ‘Adam’ optimizer was used to minimize the loss. We started with the initial learning rate of 0.0005 and gradually decrease it towards later part of training. The maximum number of epochs were set to 1000. However, the performance of validation set after each epoch of training was considered as the early termination criteria. We implemented the model using deep learning library ‘Tensorflow’.

III. Result

We evaluate the performance of the proposed CNN model on a test set consisting 1937 images distributed among 10 classes. It was made sure that none of the test dataset image was the part of training and validation dataset. Figure 3 shows the confusion matrix obtained for the test dataset. The average accuracy on a test dataset was observed to be around 96%. The test dataset was captured in various illumination conditions and having various poses and facial expressions. Even with such large variations, the proposed CNN model achieves satisfactory performance. The high accuracy achieved validate the discriminant feature extraction capability of CNN and also shows the robustness of CNN to varying conditions in the input data.

IV. Conclusion

In this work, we proposed a CNN model for face recognition. For training purpose, we created our own dataset consisting of face images of 10 different subjects. Our dataset includes face images with varying illumination, pose and facial expression to make the model more robust to changes in the input. Even with large variation in test subject the proposed

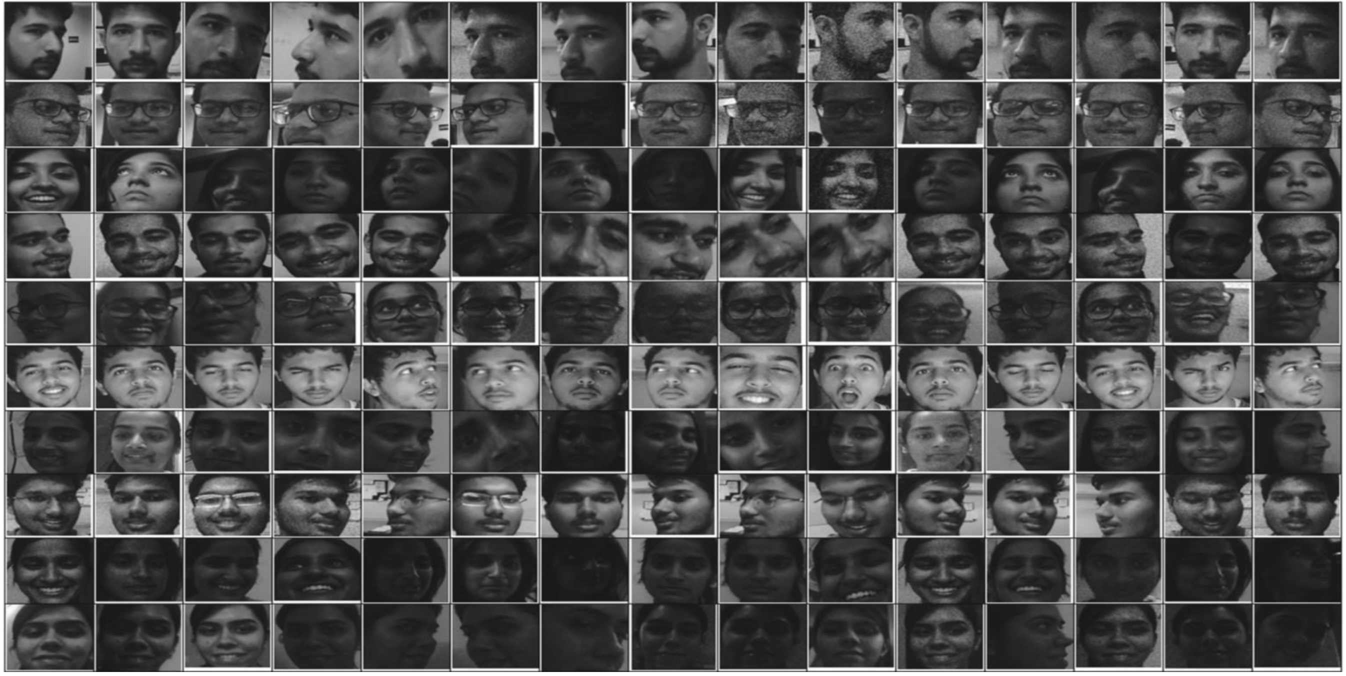


Fig. 1. Sample dataset of 10 classes

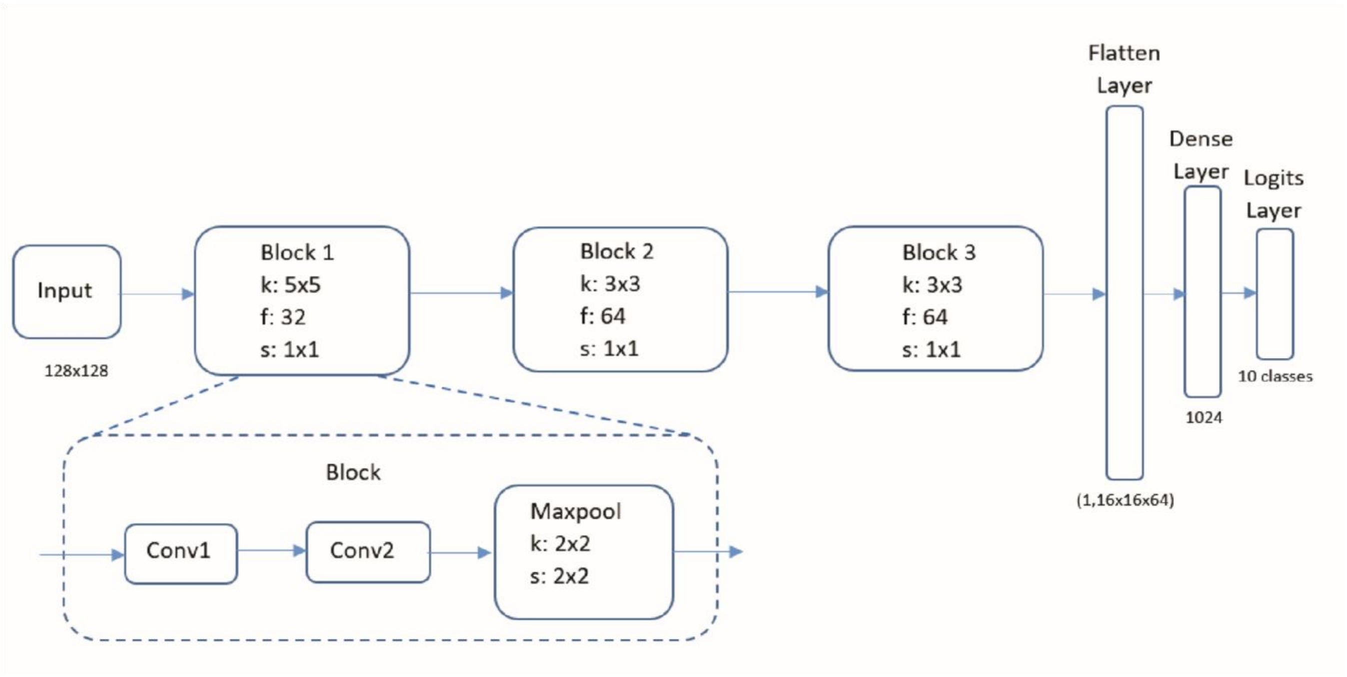
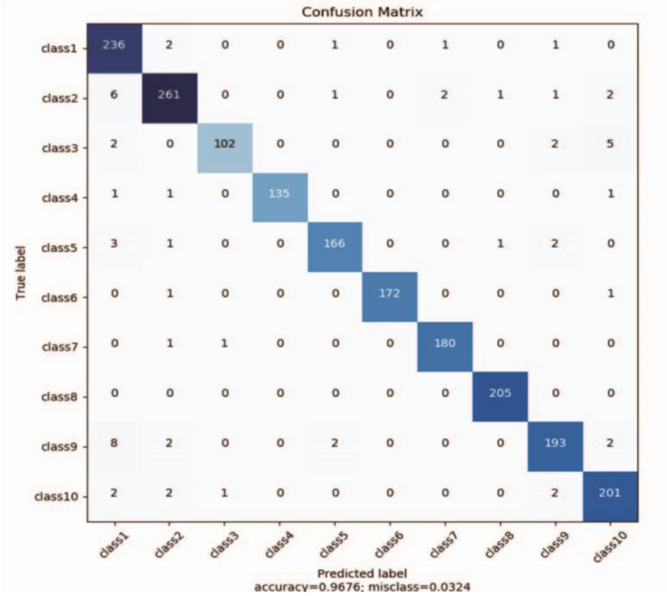


Fig. 2. Schematic of the proposed CNN architecture [k: kernel size, f: feature maps, s: stride]



	class1	class2	class3	class4	class5	class6	class7	class8	class9	class10
class1	236	2	0	0	1	0	1	0	1	0
class2	6	261	0	0	1	0	2	1	1	2
class3	2	0	102	0	0	0	0	0	2	5
class4	1	1	0	135	0	0	0	0	0	1
class5	3	1	0	0	166	0	0	1	2	0
class6	0	1	0	0	0	172	0	0	0	1
class7	0	1	1	0	0	0	180	0	0	0
class8	0	0	0	0	0	0	0	205	0	0
class9	8	2	0	0	2	0	0	0	193	2
class10	2	2	1	0	0	0	0	0	2	201

Fig. 3. Confusion matrix for test dataset.

model achieved good accuracy in classifying the facial images of different subjects. However, the proposed model was tested on lower number of classes. Creating the dataset with large number of subjects and testing the performance on large scale dataset remains as a future work.

REFERENCES

[1] R. S. El-Sayed, A. El Kholy, and M. El-Nahas, “Robust facial expression recognition via sparse representation and multiple gabor filters,” *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 3, 2013.

[2] J. H. Chen and H. P. Huang, “Face recognition using aam and global shape features,” in *2008 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2009, pp. 824–827.

[3] M. Yang and L. Zhang, “Gabor feature based sparse representation for face recognition with gabor occlusion dictionary,” in *European conference on computer vision*. Springer, 2010, pp. 448–461.

[4] J. Yang, D. D. Zhang, A. F. Frangi, and J.-y. Yang, “Two-dimensional pca: a new approach to appearance-based face representation and recognition,” *IEEE transactions on pattern analysis and machine intelligence*, 2004.

[5] J. Yang, H. Yu, and W. Kunz, “An efficient lda algorithm for face recognition,” in *Proceedings of the International Conference on Automation, Robotics, and Computer Vision (ICARCV 2000)*, 2000, pp. 34–47.

[6] T. Ahonen, A. Hadid, and M. Pietikainen, “Face recognition with local¨ binary patterns,” in *European conference on computer vision*. Springer, 2004, pp. 469–481.

[7] Y. Li, Z. Ou, and G. Wang, “Face recognition using gabor features and support vector machines,” in *International Conference on Natural Computation*. Springer, 2005, pp. 119–122.

[8] Q. Liu, H. Lu, and S. Ma, “Improving kernel fisher discriminant analysis for face recognition,” *IEEE transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 42–49, 2004.

[9] G. Du, F. Su, and A. Cai, “Face recognition using surf features,” in *MIPPR 2009: Pattern Recognition and Computer Vision*, vol. 7496. International Society for Optics and Photonics, 2009, p. 749628.

[10] Y. Xu, Q. Zhu, Z. Fan, M. Qiu, Y. Chen, and H. Liu, “Coarse to fine k nearest neighbor classifier,” *Pattern recognition letters*, vol. 34, no. 9, pp. 980–986, 2013.

[11] C. MageshKumar, R. Thiyagarajan, S. Natarajan, S. Arulselvi, and G. Sainarayanan, “Gabor features and lda based face recognition with ann classifier,” in *2011 International Conference on Emerging Trends in Electrical and Computer Technology*. IEEE, 2011, pp. 831–836.

[12] G. Guo, S. Z. Li, and K. Chan, “Face recognition by support vector machines,” in *Proceedings fourth IEEE international conference on automatic face and gesture recognition (cat. no. PR00580)*. IEEE, 2000, pp. 196–201.

[13] V. V. Kohir and U. B. Desai, “Face recognition using a dct-hmm approach,” in *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV’98 (Cat. No. 98EX201)*. IEEE, 1998, pp. 226–231.

[14] M. H. Hassoun *et al.*, *Fundamentals of artificial neural networks*. MIT press, 1995.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.