Face Recognition Using Convolutional Neural Network and Simple Logistic Classifier

Hurieh Khalajzadeh, Mohammad Mansouri and Mohammad Teshnehlab

Abstract In this paper, a hybrid system is presented in which a convolutional neural network (CNN) and a Logistic regression classifier (LRC) are combined. A CNN is trained to detect and recognize face images, and a LRC is used to classify the features learned by the convolutional network. Applying feature extraction using CNN to normalized data causes the system to cope with faces subject to pose and lighting variations. LRC which is a discriminative classifier is used to classify the extracted features of face images. Discriminant analysis is more efficient when the normality assumptions are satisfied. The comprehensive experiments completed on Yale face database shows improved classification rates in smaller amount of time.

1 Introduction

Effective methods in the extraction of features and classification methods for the extracted features are the key factors in many real-world pattern recognition and classification tasks [1]. Neural networks such as multilayer perceptron (MLP) are considered as one of the simplest classifiers that can learn from examples. An MLP can approximate any continuous function on a compact subset to any desired accuracy [2].

H. Khalajzadeh (✉) · M. Mansouri · M. Teshnehlab
K. N. Toosi University of Technology, Tehran, Iran
e-mail: hurieh.khalajzadeh@gmail.com

M. Mansouri
e-mail: mohammad.mansouri@ee.kntu.ac.ir

M. Teshnehlab
e-mail: teshnehlab@eetd.kntu.ac.ir

The performance of such classifiers depends strongly on an appropriate preprocessing of the input data [3]. In the traditional models a hand designed feature extractor congregates relevant information from the input. Designing the feature extractor by hand requires a lot of heuristics and, most notably, a great deal of time [4]. Moreover, it is not apparent what illustrates an optimal preprocessing or if there even exists an optimal solution. Sometimes it is required the input dimension to be reduced as far as possible [3]. The main deficiency of MLP neural networks for high dimensional applications such as image or speech recognition is that they offered little or no invariance to translations, shifting, scaling, rotation, and local distortions of the inputs [4].

Convolutional neural networks [5] were proposed to address all problems of simple neural networks such as MLPs. CNNs are feed-forward networks with the ability of extracting topological properties from the unprocessed input image without any preprocessing needed. Thus, these networks integrate feature selection into the training process [3]. Furthermore, CNNs can recognize patterns with extreme variability, with robustness to distortions and simple geometric transformations like translation, scaling, rotation, squeezing, stroke width and noise [6, 7]. Different versions of convolutional neural networks are proposed in the literature.

In this paper, two types of discriminative learning methods: Logistic regression classifier and Convolutional Networks are combined. LRCs [8, 9] could recognize unseen data in same accuracy as popular discriminative learning methods such as support vector machines (SVMs) [10] in a very significant less time. To investigate its efficiency and learning capability, a training algorithm is developed using back propagation gradient descent technique. The proposed architecture was tested by training the network to recognize faces.

The rest of this paper is organized as follows. Section 2 discusses an introduction to CNNs and Logistic Regression method. The proposed combined structure base on CNN and LRC is outlined in Sect. 3. The Yale dataset [11] which is used in experiments is introduced in Sect. 4. Section 5 reports the simulation results completed on the introduced database. Finally, in Sect. 6 conclusive remarks are resumed.

2 Logistic Regression, Convolutional Neural Networks

In this section, we briefly describe the logistic regression method and the Convolutional neural networks. The focus is to inspect their internal structures to provide insights into their respective strengths and weaknesses on the present vision task. This analysis will lead us to propose a method that combines the strengths of the two methods.

2.1 Convolution Neural Networks

Yann LeCun and Yoshua Bengio introduced the concept of CNNs in 1995. A convolutional neural network is a feed-forward network with the ability of extracting topological properties from the input image. It extracts features from the raw image and then a classifier classifies extracted features. CNNs are invariance to distortions and simple geometric transformations like translation, scaling, rotation and squeezing.

Convolutional neural networks combine three architectural ideas to ensure some degree of shift, scale, and distortion invariance: local receptive fields, shared weights, and spatial or temporal sub-sampling [5]. The network is usually trained like a standard neural network by back propagation.

CNN layers alternate between convolution layers with feature map $C^i_{k,l}$ shown by (1),

$$C^i_{k,l} = g(I^i_{k,l} \otimes W_{k,l} + B_{k,l}) \tag{1}$$

And non-overlapping sub-sampling layers with feature map $S^i_{k,l}$ shown by (2),

$$S^i_{k,l} = g(I^i_{k,l} \downarrow W_{k,l} + Eb_{k,l}) \tag{2}$$

Where g(x) = tanh(x) is a sigmoidal activation function, B and b the biases, W and w the weights, $I^i_{k,l}$ the i'th input and $\downarrow$ the down-sampling symbol. E is a matrix whose elements are all one and $\otimes$ denotes a 2-dimensional convolution. Note that upper case letters symbolize matrices, while lower case letters present scalars [12].

A convolutional layer is used to extract features from local receptive fields in the preceding layer. In order to extract different types of local features, a convolutional layer is organized in planes of neurons called feature maps which are responsible to detect a specific feature. In a network with a 5×5 convolution kernel each unit has 25 inputs connected to a 5×5 area in the previous layer, which is the local receptive field. A trainable weight is assigned to each connection, but all units of one feature map share the same weights. This feature which allows reducing the number of trainable parameters is called weight sharing technique and is applied in all CNN layers. LeNet5 [5], a fundamental model of CNNs proposed by LeCun, has only 60,000 trainable parameters out of 345,308 connections. A reduction of the resolution of the feature maps is performed through the subsampling layers. In a network with a 2×2 subsampling filter such a layer comprises as many feature map numbers as the previous convolutional layer but with half the number of rows and columns.

In the rest of this section LeNet5 which is a particular convolutional neural network is described. LeNet5 takes a raw image of 32×32 pixels as input. It is composed of seven layers: three convolutional layers (C1, C3 and C5), two sub-sampling layers (S2 and S4), one fully connected layer (F6) and the output layer. The output layer is an Euclidean RBF layer of 10 units (for the 10 classes) [13]. These layers are connected as shown in Fig. 1.

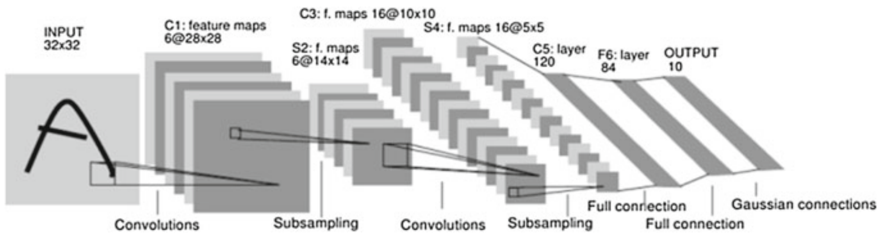

Fig. 1 LeNet5 architecture [5]

Table 1 The interconnection of the S2 layer to C3 layer [5]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	x				x	x	x			x	x	x	x		x	x
1	x	x				x	x	x			x	x	x	x		x
2	x	x	x				x	x	x			x		x	x	x
3		x	x	x			x	x	x	x			x		x	x
4			x	x	x			x	x	x	x		x	x		x
5				x	x	x			x	x	x	x		x	x	x

As shown in Table 1 the choice is made not to connect every feature map of S2 to every feature map of C3. Each unit of C3 is connected to several receptive fields at identical locations in a subset of feature maps of S2 [5, 13].

2.2 *Logistic Regression Classifier*

Logistic Regression is an approach to learning functions of the form $f : X \rightarrow Y$, or $P(Y|X)$ in the case where Y is discrete-valued, and $X = <X_1 \cdots X_n>$ is any vector containing discrete or continuous variables. In this section we will primarily consider the case where Y is a boolean variable, in order to simplify notation. More generally, Y can take on any of the discrete values $y = \{y_1 \cdots y_k\}$ which is used in experiments [9].

Logistic Regression assumes a parametric form for the distribution $P(Y|X)$, then directly estimates its parameters from the training data. The parametric model assumed by Logistic Regression in the case where Y is boolean is:

$$P(Y = 1|X) = \frac{1}{1 + exp(w_0 + \sum_{i=1}^{n} w_i X_i)} \tag{3}$$

and

$$P(Y = 0|X) = \frac{exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + exp(w_0 + \sum_{i=1}^{n} w_i X_i)} \tag{4}$$

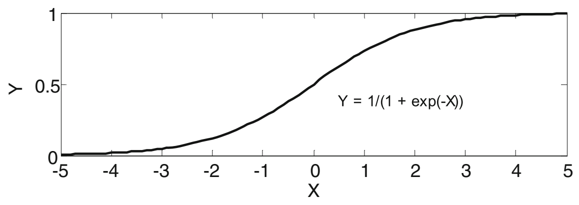

Fig. 2 Form of the logistic function. $P(Y|X)$ is assumed to follow this form [9]

Notice that Eq. (4) follows directly from Eq. (3), because the sum of these two probabilities must equal 1 [9].

One highly expedient property of this form for $P(Y|X)$ is that it leads to a simple linear expression for classification. To classify any given X we generally want to assign the value y_k that maximizes $P(Y = y_k|X)$. Put another way, we assign the label $Y = 0$ if the following condition holds [9]:

$$1 < \frac{P(Y = 0|X)}{P(Y = 1|X)}$$

Substituting from Eqs. (3) and (4), this becomes

$$1 < exp(w_o + \sum_{i=1}^{n} w_i X_i)$$

Form of the logistic function is shown in Fig. 2.

And taking the natural log of both sides, we have a linear classification rule that assigns label $Y = 0$ if X satisfies

$$0 < w_o + \sum\nolimits_{i=1}^{n} w_i X_i, \tag{5}$$

And assigns $Y = 1$ otherwise [9].

3 Proposed Method

In this paper, a CNN structure depicted in Fig. 3 is considered for feature extraction. The applied CNN is composed of four layers: input layer, two convolutional layers and one subsampling layer. Size of input layer is considered as 64×64, so the input images of the applied database were resized to 64×64 to be compatible with the proposed structure. The first convolutional layer has six feature maps, each of which has resolution of 58×58, with a receptive field of 7×7. The second layer, or the subsampling layer, contains six feature maps of size 29×29, with a receptive field of 2×2. The second convolutional layer has sixteen feature maps, each of which has resolution of 22×22, with a receptive field of 8×8. The second subsampling layer contains sixteen feature maps of size 11×11, with a receptive field of 2×2. The output layer is a fully connected layer with 15 feature maps with the size of 1×1, with a receptive field of 11×11. The interconnections between first Subsampling layer and second convolutional layer are shown in Table 2.

Table 2 Interconnection of first subsampling layer with the second convolutional layer

	0	1	2	3	4	5	6	7	8	9
0	x				x	x	x			
1	x	x				x	x	x		
2	x	x	x				x	x	x	
3		x	x	x				x	x	x
4			x	x	x				x	x
5				x	x	x				x

The primary weights and biases are considered in the range of -0.5 and 0.5.

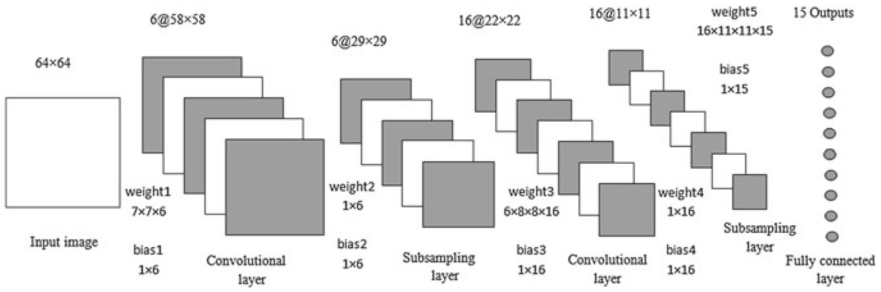

Fig. 3 CNN structure used for feature extraction

Due to computational complexity and time limitation of CNNs, we considered 500 epochs for the comparisons. At first, learning rate is considered the same as [7]:

$$\eta = \frac{\eta_0}{\frac{n}{N/2} + \frac{c_1}{max(1,(c_1 \frac{max(0,c_1(n-c_2N))}{(1-c_2)N}))}} \tag{6}$$

Where η = learning rate, $\eta_0 =$ initial learning rate = 0.1, N = total training epochs, n = current training epoch, $c_1 = 50$ and $c_2 = 0.65$. Learning rate as a function of epoch number is shown in Fig. 4.

Using the mentioned learning rate, after passing approximately half of the maximum number of epochs, no more improvement was resulted for accuracy and error rate. Therefore, a new learning rate is created by a little alteration in the Eq. (6) which is shown in Eq. (7). In the proposed learning rate, after passing 0.65% of epochs a small value (0.001) is replaced. It causes an abrupt change in both accuracy and error rates. Keeping the learning rate as a constant value leads to increase exploitation and to yield the convergence of learning algorithm sooner.

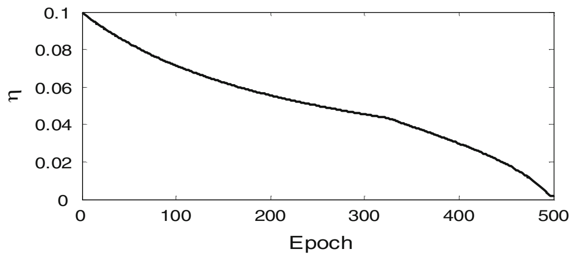

Fig. 4 Learning rate (η_1) [7]

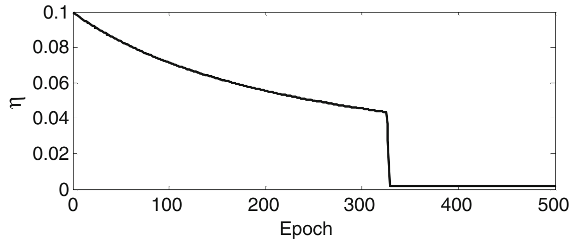

Fig. 5 Learning rate (η_2)

$$\eta = \frac{\eta_0}{\frac{n}{N/2} + \frac{c_1}{max(1,(c_1 \frac{max(0,c_1^2(n-c_2N))}{(1-c_2)N}))}} \tag{7}$$

The new learning rate is shown in Fig. 5. The comparisons based on the two learning rate are brought in the following.

Images are normalized to lie in the range -1 to 1 by removing the mean value and dividing them by their standard deviation as shown in (8).

$$I_n = \frac{I - \bar{I}}{\sigma} \tag{8}$$

Where I is the input image, $\bar{I}$ is the mean of all input image pixels, σ is the standard deviation and I_n is the normalized image.

By applying CNNs to a small database, number of trainable parameters surpasses the number of data. For this reason, weights in the network were updated sequentially after each pattern presentation, as opposed to batch update that weights are only updated once per pass through the training set.

A simple back-propagation gradient descent algorithm without any optimization is applied for training the network. Optimizing the network is postponed to the future works. Error function is assumed to be the squared error,

$$E = \frac{1}{2}e^2 = \frac{1}{2}(t - y)^2 \tag{9}$$

Where t is the target output, y is the actual network output, and e is the network error.

4 Dataset

To evaluate the performance of the proposed structure, various simulations are performed on a commonly used face database. Some samples of this database are shown in Fig. 6.

The Yale face database contains 165 face images of 15 individuals. There are 11 images per subject, one for each facial expression or configuration: center-light, glasses/no glasses, happy, normal, left-light, right-light, sad, sleepy, surprised and wink. Each image was digitized and presented by a 61×80 pixel array whose gray levels ranged between 0 and 255.

Table 3 summarizes what the number of data and classes in the database is. Also presented are train and test data partitioning for database images.

5 Simulation Results and Discussions

Firstly, classifying the extracted features is done using a winner takes all mechanism. There are 15 output neurons extracted for each image data. The biggest output neuron specified the class of the data. The recognition accuracy, training time and number of trainable parameters are summarized in Table 4. Also, train and test accuracy is shown in Fig. 7.

Several machine learning algorithms are applied on the features extracted from the database images using CNN. Machine learning algorithms are simulated using WEKA software. Results with these methods are reported and compared in Tables 5 and 6.

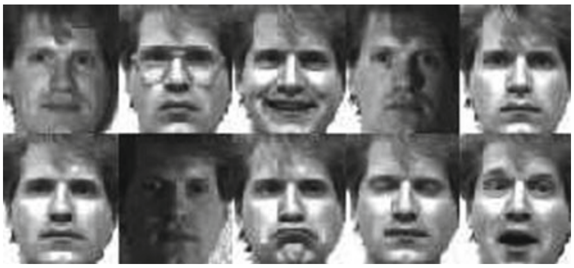
Fig. 6 Samples of Yale face database images

Table 3 Number of data and classes in database

Number of data	Number of train data	Number of test data	Classes	Train sample per class	Test sample per class
165	135	30	15	9	2

Table 4 Recognition accuracy, training time and number of parameters for Yale database

Number of parameters	Average training per epoch time (s)	Train accuracy (%)	Test accuracy (%)
35559	65	88.88	80

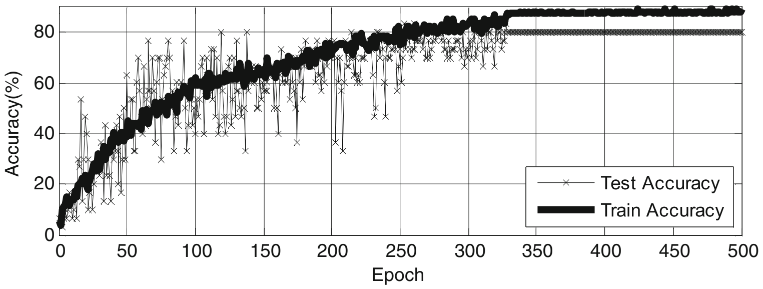

Fig. 7 Train and test data classification accuracy on Yale well-known face data set

Three different methods of Naïve Bayes classifier consist of NaiveBayes, NaiveBayesSimple and NaiveBayesUpdateable, functional classifiers consist of Logistic, MultiLayerPerceptron, RBFNetwork, SimpleLogistic and SOM which is shown the Support Vector Machine, lazy methods consist of IB1, IBk and LWL (Local Weighted Learning), Classification via regression methods and trees consist of BFTree, FT, LMT and SimpleCart are tested on feature vectors which are resulted using CNN. IBk method is repeated for k in the range of 1 to 10. The best accuracy is resulted by k = 7 which is shown in Table 5.

As it is shown, experimental results show that Simple Logistic algorithm could recognize the unseen face data with the highest classification accuracy of 86.06 in the least time in compare with other algorithms. LMT tree is also resulted the same classification accuracy in a bit more time. Classification accuracy and classification time on the Yale dataset for various classification methods are shown and compared in Fig. 8.

Table 5 Results of IBk algorithm

k	1	2	3	4	5	6	7	8	9	10
Accuracy(%)	76.36	79.39	81.21	81.81	81.81	83.03	83.63	83.03	83.03	83.03

Table 6 Comparison of different algorithms

	Accuracy (%)	Time (s)
NaiveBayes	75.15	0.05
NaiveBayesSimple	75.15	0.01
NaiveBayesUpdateable	75.15	0
Logistic	65.45	1.23
MultiLayerPerceptron	81.21	2.88
RBFNetwork	72.72	23
SimpleLogistic	86.06	1.22
SupportVectorMachine	83.03	9.79
IB1	76.36	0
IBk	83.63	0
LWL	78.78	0
ClassificationViaRegression	81.21	0.28
BFTree	75.15	0.21
FT	71.51	0.64
LMT	86.06	1.64
SimpleCart	75.75	0.22

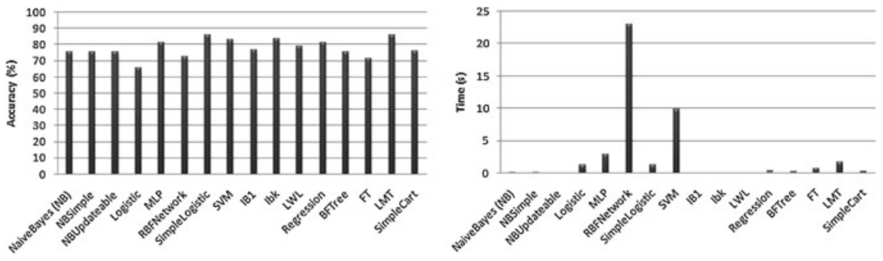

Fig. 8 Classification accuracy and time on the test set for various classification methods

6 Conclusion

In this paper, convolutional neural networks and simple logistic regression method are investigated with results on Yale face dataset. To combine the advantages of the two methods a two step learning process is proposed: first, training a convolutional neural network and view the first N – 1 layers as a feature extractor. Second, applying a simple logistic regression method on the features produced by the convolutional neural network. Proposed structure benefits from all CNN advantages such as feature extracting and robustness to distortions. The network was trained using back-propagation gradient descent algorithm. Applicability of these networks in recognizing face images is presented in this paper. Based on experiments on the Yale dataset, feature extraction using CNN which is applied to normalized data causes the system to cope with faces subject to pose and lighting variations. Moreover, simple logistic regression method classifies the extracted features with the highest classification accuracy and lowest classification time in compare with other machine learning algorithms.

References

1. Lo, S.B., Li, H., Wang, Y., Kinnard, L., Freedman, M.T.: A multiple circular path convolution neural network system for detection of mammographic masses. IEEE Trans. Med. Imaging 150–158 (2002)
2. Hu, Y.H., Hwang, J.: Handbook of neural network signal processing. CRC Press, Boca Raton (2002)
3. Gepperth, A.: Object detection and feature base learning by sparse convolutional neural networks. In: Lecture notes in artificial intelligence 4807, Springer Verlag Berlin, 221–231 (2006)
4. Bouchain, D.: Character Recognition Using Convolutional Neural Networks. In: Seminar Statistical Learning Theory, University of Ulm, Germany (2006/2007)
5. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. vol. 86, pp. 2278–2324 (1998)
6. Ahranjany, S.S., Razzazi, F., Ghassemian, M.H.: A very high accuracy handwritten character recognition system for Farsi/Arabic digits using Convolutional Neural Networks. In: IEEE Fifth International Conference on Bio-Inspired Computing, Theories and Applications (BIC-TA) pp. 1585–1592 (2010)
7. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural network approach. In: IEEE Trans. Neural Networks 98–113 (1997)
8. Palei, S.K., Das, S.K.: Logistic regression model for prediction of roof fall risks in bord and pillar workings in coal mines: An approach. Saf. Sci. (2009)
9. Mitchell, T.M.: Generative and discriminative classifiers: naive bayes and logistic regression. Machine Learning (2010)
10. LeCun, Y., Huang, F.-J.: Large-scale learning with SVM and convolutional nets for generic object categorization. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'06). IEEE Press, Salt Lake City (2006)
11. Yale face database, http://cvc.yale.edu/projects/yalefaces/yalefaces.html
12. Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J.: Coding facial expressions with gabor wavelets. In: Third IEEE International Conference on Automatic Face and Gesture Recognition pp. 200–205 (1998)
13. Fasel, B.: Robust face analysis using convolutional neural networks. In: Proceedings of the 16th International Conference on Pattern Recognition, pp. 40–43 (2002)