

# A Correlational Neural Network for Gender Classification

Ting Zhang<sup>a</sup>, Yujian Li<sup>b\*</sup>, Haihe Hu<sup>c</sup>, Yahong Zhang<sup>d</sup>

School of Computer Science

Beijing University of Technology, Beijing 100124, China

<sup>a</sup> zhangting08@emails.bjut.edu.cn; {<sup>b</sup> liyujian, <sup>c</sup> huhaihe}@bjut.edu.cn; <sup>d</sup> plahpu@163.com

\*corresponding author

**Keywords:** Gender classification, convolutional neural network, correlational neural network, correlational operation.

**Abstract.** A convolutional neural network (CNN) can perform well in a variety of applications such as human face gender classification, but requiring flips of convolutional kernels in implementation. By replacing convolution with correlation, we propose a correlational neural network (CorNN) instead of a CNN. A CorNN takes advantage over a CNN in that it requires no flips of correlational kernels in implementation, saving a lot of training and testing time. Experimental results show that an 8-layer CorNN for gender classification can not only perform as well as the corresponding CNN, but also run surprisingly faster with a relative reduction of 11.29%~18.83% training time, and 10.16%~16.57% testing time.

## 1. Introduction

Human face gender classification is a subject of automatically identifying the gender (male or female) of a human face image. It is easy for a human but challenging for a computer [1], with wide applications in customer advertisement, visual surveillance, intelligent interface, population statistics, and so on.

As a traditional method, fully-connected neural networks were applied to identification of human face gender many years ago [2] [3][4][5]. However, the fully-connected neural networks did not take into account the two-dimensional structure of an image when extracting features from it. As a consequence, they are not satisfactory in terms of gender classification accuracy. Their drawbacks ignoring 2D structure information could be overcome partly by convolutional neural networks (CNNs). Recently, Verma et al. have implemented a 6-layer CNN to identify face gender. However, this CNN needs plenty of epochs which is 25000 to achieve 88.46% accuracy on a dataset containing 4700 face images from the web [6].

Primarily, a CNN is designed to recognize 2D shapes by inspiration of the visual neural mechanism with some degree of translation invariance. In 1962, Hubel and Wiesel proposed the concept of receptive field in studying the cat's visual cortex cells [7]. With the insight of receptive field, Fukushima developed the model of neocognitron [8], perhaps the first implemented CNN.

Commonly, a CNN is composed of input layer, convolutional layer, subsampling layer, convolutional layer, subsampling layer, ..., fully-connected layers and output layer. It has two distinguishing features: local connectivity and shared weights. The local connectivity allows the network to first create good representations of small parts of the input, then assemble representations of larger areas from them. The shared weights allow for features to be detected regardless of their position in the visual field, thus constituting the property of translation invariance.

Due to the superior ability of feature extraction, CNNs have been successfully applied to many important fields, such as character recognition [9], face recognition [10], face tracking [11], and traffic sign recognition [12]. Especially in ImageNet Large Scale Visual Recognition Challenge (ILSVRC), CNNs play a vital role in achieving the best performance [13].

However, a CNN has a disadvantage in implementation of convolutional operation, which may require a huge number of horizontal flips and vertical flips of convolutional kernels. This can obviously slow down the training speed of a CNN on big datasets. In order to eliminate the flipping effect, we present a new model, called correlational neural network (CorNN). Compared to a CNN, a CorNN has no requirement of flipping convolutional kernels, thus running faster than a CNN. Furthermore, we implement an 8-layer CorNN for gender classification in this paper.

The rest of the paper is organized as follows. Section 2 gives the model of CorNN in detail. Section 3 makes a comparative evaluation for gender classification of CorNN and CNN in experiments. Finally, Section 4 makes a few conclusions.

## 2. Correlational Neural Networks

### 2.1. Equivalence between CNNs and CorNNs

A CNN has four basic operations: inner convolution, outer convolution, subsampling, and upsampling. These operations are defined as follows.

Suppose  $\mathbf{A}$  is a  $M \times N$  matrix and  $\mathbf{B}$  is a  $m \times n$  matrix. If  $M \geq m$  and  $N \geq n$ , then their inner convolution  $\mathbf{C} = \mathbf{A} * \mathbf{B}$  is defined with each element computed as:

$$c_{ij} = \sum_{s=1}^m \sum_{t=1}^n a_{i+s-1, j+t-1} \cdot b_{m-s+1, n-t+1}, \quad 1 \leq i \leq M - m + 1, 1 \leq j \leq N - n + 1. \quad (1)$$

And their outer convolution is defined by:

$$\mathbf{A} * \mathbf{B} = \widehat{\mathbf{A}}_{\mathbf{B}} * \mathbf{B}, \quad (2)$$

where  $\widehat{\mathbf{A}}_{\mathbf{B}} = (\widehat{a}_{ij})$  is a matrix of size  $(M + 2m - 2) \times (N + 2n - 2)$  that extends  $\mathbf{A}$  with 0, namely,

$$\widehat{a}_{ij} = \begin{cases} a_{i-m+1, j-n+1}, & \text{if } m \leq i \leq M + m - 1 \text{ and } n \leq j \leq N + n - 1 \\ 0, & \text{otherwise} \end{cases}. \quad (3)$$

If matrix  $\mathbf{A}$  is divided into non-overlapping blocks of size  $\lambda \times \tau$  with the  $ij$ -th block denoted by  $\mathbf{G}_{\lambda, \tau}^{\mathbf{A}}(i, j)$ , then we have

$$\mathbf{G}_{\lambda, \tau}^{\mathbf{A}}(i, j) = (a_{st})_{\lambda \times \tau}, \text{ where } (i-1) \cdot \lambda + 1 \leq s \leq i \cdot \lambda, (j-1) \cdot \tau + 1 \leq t \leq j \cdot \tau. \quad (4)$$

Here the subsampling of  $\mathbf{G}_{\lambda, \tau}^{\mathbf{A}}(i, j)$  is chosen as the mean pooling,

$$\text{down}(\mathbf{G}_{\lambda, \tau}^{\mathbf{A}}(i, j)) = \frac{1}{\lambda \times \tau} \sum_{s=(i-1)\lambda+1}^{i\lambda} \sum_{t=(j-1)\tau+1}^{j\tau} a_{st}. \quad (5)$$

The non-overlapping subsampling of matrix  $\mathbf{A}$  with block size  $\lambda \times \tau$  is defined as:

$$\text{down}_{\lambda, \tau}(\mathbf{A}) = \text{down}(\mathbf{G}_{\lambda, \tau}^{\mathbf{A}}(i, j)). \quad (6)$$

The non-overlapping upsampling of matrix  $\mathbf{A}$  with multiple size  $\lambda \times \tau$  is defined as:

$$up_{\lambda \times \tau}(\mathbf{A}) = \mathbf{A} \otimes \mathbf{1}_{\lambda \times \tau}, \quad (7)$$

where  $\mathbf{1}_{\lambda \times \tau}$  is a matrix of 1, and  $\otimes$  denotes the Kronecker product.

Correspondingly, a CorNN also has four basic operations: inner correlation, outer correlation, subsampling, and upsampling.

The definitions of subsampling and upsampling in the CorNN are the same as in the CNN.

The inner correlation  $\mathbf{C} = \mathbf{A} \star \mathbf{B}$  between matrix  $\mathbf{A}$  and matrix  $\mathbf{B}$  is defined as:

$$c_{ij} = \sum_{s=1}^m \sum_{r=1}^n a_{i+s-1, j+r-1} \cdot b_{sr}, \quad 1 \leq i \leq M-m+1, 1 \leq j \leq N-n+1. \quad (8)$$

The outer correlation between  $\mathbf{A}$  and  $\mathbf{B}$  is defined as:

$$\mathbf{A} \circ \mathbf{B} = \widehat{\mathbf{A}}_{\mathbf{B}} \star \mathbf{B}, \quad (9)$$

where  $\widehat{\mathbf{A}}_{\mathbf{B}}$  is the same as in formula (2).

For example, if  $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  and  $\mathbf{B} = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$ , then  $\widehat{\mathbf{A}}_{\mathbf{B}}$  is:

$$\widehat{\mathbf{A}}_{\mathbf{B}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 & 0 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (10)$$

Accordingly, the inner and outer convolutions of  $\mathbf{A}$  and  $\mathbf{B}$  are:

$$\mathbf{A} \star \mathbf{B} = \begin{bmatrix} 35 & 49 \\ 77 & 91 \end{bmatrix} \quad \mathbf{A} \circ \mathbf{B} = \begin{bmatrix} 2 & 7 & 12 & 9 \\ 12 & 35 & 49 & 33 \\ 30 & 77 & 91 & 57 \\ 28 & 67 & 76 & 45 \end{bmatrix} \quad (11)$$

Meanwhile, the inner and outer correlations of  $\mathbf{A}$  and  $\mathbf{B}$  are:

$$\mathbf{A} \star \mathbf{B} = \begin{bmatrix} 49 & 63 \\ 91 & 105 \end{bmatrix} \quad \mathbf{A} \circ \mathbf{B} = \begin{bmatrix} 5 & 14 & 23 & 12 \\ 23 & 49 & 63 & 30 \\ 47 & 91 & 105 & 48 \\ 21 & 38 & 43 & 18 \end{bmatrix} \quad (12)$$

Generally, the inner and outer convolutions of two given matrices are different from their inner and outer correlations. However, in theory an arbitrary CNN can be simulated by a corresponding CorNN, and vice versa.

In fact, for any convolutional kernel  $\mathbf{w}$  of size  $m \times n$ , we can define a horizontal flip of  $\mathbf{w}$  below:

$$\mathbf{w}' = HF(\mathbf{w}), \quad (13)$$

where  $w'_{ij} = w_{i, n-j+1}, 1 \leq i \leq m, 1 \leq j \leq n$ .

Moreover, we can also define a vertical flip of  $\mathbf{w}'$  below:

$$\mathbf{w}'' = VF(\mathbf{w}'), \quad (14)$$

where  $w''_{ij} = w'_{m-i+1, j}, 1 \leq i \leq m, 1 \leq j \leq n$ .

Hence, for an input (or a convolutional map)  $\mathbf{x}$ , we have

$$\mathbf{x} \star \mathbf{w} = \mathbf{x} \star VF(HF(\mathbf{w})) = \mathbf{x} \star \mathbf{w}'' \quad (15)$$

This means that if we replace all convolutional kernels  $\mathbf{w}$  in a CNN with their corresponding correlational kernels, namely,

$$\mathbf{w}^* = VF(HF(\mathbf{w})), \quad (16)$$

we can get a CorNN that always produce the same output as the CNN for the same input.

In the meantime, because  $\mathbf{w} = VF(HF(\mathbf{w}^*))$ , we can also get a corresponding CNN that always produce the same output as any given CorNN for the same input.

Therefore, CNN is equivalent to CorNN in functionality. Because a convolutional operation may require a large number of vertical and horizontal flips in implementation, it can be expected that a CorNN should be more efficient than the corresponding CNN in training and testing. Therefore, we propose a CorNN instead of a CNN for gender classification in the next subsection.

## 2.2. A CorNN for Gender Classification

As mentioned previously, a 6-layer CNN has been designed to identify face gender [6], but requiring more than twenty thousand epochs of training to achieve a satisfactory result. Considering deeper neural networks may perform better [14], in Fig.1 we present an 8-layer CorNN for gender classification. The CorNN consists of an input layer  $\mathbf{x}$ , three correlational layers ( $\mathbf{h}_1, \mathbf{h}_3, \mathbf{h}_5$ ) of sigmoid neurons, three subsampling layers ( $\mathbf{h}_2, \mathbf{h}_4, \mathbf{h}_6$ ), and an output layer  $\mathbf{o}$ . The input is a face image for correlational layers to extract gender features with subsampling layers for dimensionality reduction. The three correlational layers are ordered alternately with the three subsampling layers. The last subsampling layer is directly connected to the output layer. The output layer is composed of two sigmoid neurons for gender classification.

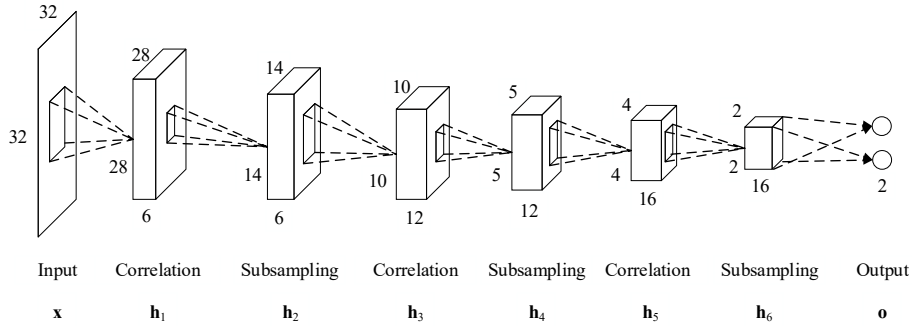


Figure 1 The architecture of the proposed CorNN.

## 2.3. Learning Algorithm for the CorNN

For the  $l$ -th sample  $\mathbf{x}^l$ , we can describe the computing procedure of the CorNN as follows:

$$\begin{cases}
\mathbf{h}_{1,j}^l = \sigma(\mathbf{u}_{1,j}^l) = \sigma(\mathbf{x}^l \tilde{\mathbf{w}}^{1,j} + \mathbf{b}^{1,j}), 1 \leq j \leq 6 \\
\mathbf{h}_{2,j}^l = g(\mathbf{u}_{2,j}^l) = g(\text{down}_{\lambda,\tau}(\mathbf{h}_{1,j}^l)), 1 \leq j \leq 6 \\
\mathbf{h}_{3,j}^l = \sigma(\mathbf{u}_{3,j}^l) = \sigma\left(\sum_{i=1}^6 \mathbf{h}_{2,i}^l \tilde{\mathbf{w}}^{3,j} + \mathbf{b}^{3,j}\right), 1 \leq j \leq 12 \\
\mathbf{h}_{4,j}^l = g(\mathbf{u}_{4,j}^l) = g(\text{down}_{\lambda,\tau}(\mathbf{h}_{3,j}^l)), 1 \leq j \leq 12 \\
\mathbf{h}_{5,j}^l = \sigma(\mathbf{u}_{5,j}^l) = \sigma\left(\sum_{i=1}^{12} \mathbf{h}_{4,i}^l \tilde{\mathbf{w}}^{5,j} + \mathbf{b}^{5,j}\right), 1 \leq j \leq 16 \\
\mathbf{h}_{6,j}^l = g(\mathbf{u}_{6,j}^l) = g(\text{down}_{\lambda,\tau}(\mathbf{h}_{5,j}^l)), 1 \leq j \leq 16 \\
\mathbf{H}_6^l = (\mathbf{h}_{6,1}^l, \mathbf{h}_{6,2}^l, \dots, \mathbf{h}_{6,16}^l), \\
\mathbf{o}^l = \sigma(\mathbf{u}_7^l) = \sigma(\mathbf{W}^7 \mathbf{H}_6^l + \mathbf{b}^7),
\end{cases}, \quad (17)$$

where  $\sigma(x) = 1/(1+e^{-x})$ ,  $g(x) = x$ .  $\mathbf{w}^{1,j}$ ,  $\mathbf{w}^{3,j}$  and  $\mathbf{w}^{5,j}$  stand for the correlational kernels (or weights) of the first, third and the fifth hidden layers,  $\mathbf{b}^{1,j}$ ,  $\mathbf{b}^{3,j}$  and  $\mathbf{b}^{5,j}$  for the related correlational bias,  $\mathbf{h}_{1,j}^l$ ,  $\mathbf{h}_{3,j}^l$  and  $\mathbf{h}_{5,j}^l$  for the related correlational maps.  $\mathbf{h}_{2,j}^l$ ,  $\mathbf{h}_{4,j}^l$  and  $\mathbf{h}_{6,j}^l$  are the corresponding subsampling maps computed from  $\mathbf{h}_{1,j}^l$ ,  $\mathbf{h}_{3,j}^l$  and  $\mathbf{h}_{5,j}^l$ , respectively.  $\mathbf{w}^7$  and  $\mathbf{b}^7$  are the weight vector and the bias vector of the output layer.  $\mathbf{o}^l$  is the output vector. Note that  $\mathbf{H}_6^l$  should be viewed as a vector.

Based on the BP learning of CNNs, it is not difficult to derive a BP algorithm for CorNNs, called CorNN-BP, to learn the parameters of the 8-layer CorNN discussed in Section 2.2.

Suppose we have  $N$  two-class training samples  $(\mathbf{x}^l, \mathbf{y}^l)$  ( $1 \leq l \leq N$ ), where  $\mathbf{y}^l$  is the expected output for the input  $\mathbf{x}^l$ , with the actual output  $\mathbf{o}^l$ . The objective function is the squared error over all training samples, namely:

$$L_N = \frac{1}{2} \sum_{l=1}^N \|\mathbf{o}^l - \mathbf{y}^l\|^2. \quad (18)$$

An important part of the CorNN-BP is to compute all the sensitivities  $\delta_k$  ( $1 \leq k \leq 7$ ) as follows,

$$\begin{cases}
\delta_7^l = (\mathbf{o}^l - \mathbf{y}^l) \circ \sigma'(\mathbf{u}_7^l) \\
\delta_6^l = [(\mathbf{W}^7)^T \delta_7^l] \circ \mathbf{H}_6^l \circ (1 - \mathbf{H}_6^l) \\
\delta_6^l = [\delta_{6,1}^l, \delta_{6,2}^l, \dots, \delta_{6,j}^l, \dots, \delta_{6,16}^l], 1 \leq j \leq 16 \\
\delta_{5,j}^l = \frac{1}{\lambda \times \tau} (\sigma'(\mathbf{u}_{5,j}^l) \circ \text{up}_{\lambda \times \tau}(\delta_{6,j}^l)), 1 \leq j \leq 16 \\
\delta_{4,j}^l = [\delta_{5,j}^l \tilde{\mathbf{w}}^{5,j}] \circ g'(\mathbf{u}_{4,j}^l), 1 \leq j \leq 12 \\
\delta_{3,j}^l = \frac{1}{\lambda \times \tau} (\sigma'(\mathbf{u}_{4,j}^l) \circ \text{up}_{\lambda \times \tau}(\delta_{4,j}^l)), 1 \leq j \leq 12 \\
\delta_{2,j}^l = [\delta_{3,j}^l \tilde{\mathbf{w}}^{3,j}] \circ g'(\mathbf{u}_{2,j}^l), 1 \leq j \leq 6 \\
\delta_{1,j}^l = \frac{1}{\lambda \times \tau} (\sigma'(\mathbf{u}_{1,j}^l) \circ \text{up}_{\lambda \times \tau}(\delta_{2,j}^l)), 1 \leq j \leq 6
\end{cases}, \quad (19)$$

where ' $\circ$ ' denotes the element-wise product,  $\text{up}_{\lambda \times \tau}(\cdot)$  denotes the upsampling function defined by (7).  $\delta_7^l$  stands for the sensitivity of the output layer,  $\delta_{6,j}^l$ ,  $\delta_{5,j}^l$ ,  $\delta_{4,j}^l$ ,  $\delta_{3,j}^l$ ,  $\delta_{2,j}^l$ , and  $\delta_{1,j}^l$  for the sensitivities of the lower layers.

Using the sensitivities, the CorNN-BP then computes the partial derivatives of the objective function  $L_N$  with respect to weights and biases below,

$$\left\{ \begin{array}{l} \frac{\partial L_N}{\partial \mathbf{W}^7} = \sum_{l=1}^N \delta_l^7 (\mathbf{H}_6^l)^T, \frac{\partial L_N}{\partial \mathbf{b}^7} = \sum_{l=1}^N \delta_l^7, \\ \frac{\partial L_N}{\partial \mathbf{W}^{3,j}} = \sum_{l=1}^N \mathbf{h}_{4,j}^l \delta_{5,j}^l, \frac{\partial L_N}{\partial \mathbf{b}^{3,j}} = \sum_{l=1}^N \delta_{5,j}^l, 1 \leq j \leq 16 \\ \frac{\partial L_N}{\partial \mathbf{W}^{2,j}} = \sum_{l=1}^N \mathbf{h}_{2,j}^l \delta_{3,j}^l, \frac{\partial L_N}{\partial \mathbf{b}^{2,j}} = \sum_{l=1}^N \delta_{3,j}^l, 1 \leq j \leq 12 \\ \frac{\partial L_N}{\partial \mathbf{W}^{1,j}} = \sum_{l=1}^N \mathbf{h}_{1,j}^l \delta_{1,j}^l, \frac{\partial L_N}{\partial \mathbf{b}^{1,j}} = \sum_{l=1}^N \delta_{1,j}^l, 1 \leq j \leq 6 \end{array} \right. \quad (20)$$

Finally, the CorNN-BP updates the weights and biases using gradient descent.

The main advantage of the CorNN-BP lies in that it requires no flips of weight matrices in implementation, because of using inner correlations and outer correlations instead of inner convolutions and outer convolutions. This can save a lot of training and testing time with a bit surprise, as shown in experimental results.

### 3. Experimental Results

In order to demonstrate the validity of the proposed CorNN for face gender classification, we compare it with a CNN with the same structure on nine human face databases: ORL, Georgia Tech, FERET [15], Extended Yale B (EYB) [16], AR [17], Faces94, LFW [18], MORPH and CelebFaces+ [19]. The examples of face images from these databases are shown in Fig. 2. The proposed CorNN and the corresponding CNN are implemented in Matlab R2010a. Both of them take the learning rate of 0.1 on all the databases, with the maxepoch of 50 on the first six and 1000 on the last three. All experiments were conducted on a PC platform with i7-3770 3.10-GHz processor, 8-GB memory, and Windows 7.0 operation system.

#### 3.1. Preprocessing of Images in Databases

The nine face databases for experiments are detailed in Table 1. “MR”, “FR”, and “TR” denote number of males training samples, number of female training samples and number of total training samples, respectively. “ME”, “FE”, and “TE” denote number of male testing samples, number of female testing samples and number of total testing samples, respectively. Moreover, no person has some of his/her images in training set and others in testing set simultaneously. Additionally, ORL, FERET, EYB and AR consist of gray images, whereas Georgia Tech, Faces94, LFW, MORPH and CelebFaces+ consist of color images. Before experiments, all these images were resized to 32×32 with color images converted into gray ones, with the value of each pixel normalized into [0, 1].

Table 1 Databases used in experiments

Databases	Training Samples			Testing Samples		
	MR	FR	TR	ME	FE	TE
ORL	320	30	350	40	10	50
Georgia Tech	450	75	525	195	30	225
FERET	658	532	1190	105	105	210
EYB	1280	384	1664	576	192	768
AR	910	910	1820	390	390	780
Faces94	2000	400	2400	660	20	680
LFW	8000	1900	9900	2000	800	2800
MORPH	40997	7102	48099	3000	1000	4000
CelebFaces+	27887	37113	65000	2500	2500	5000

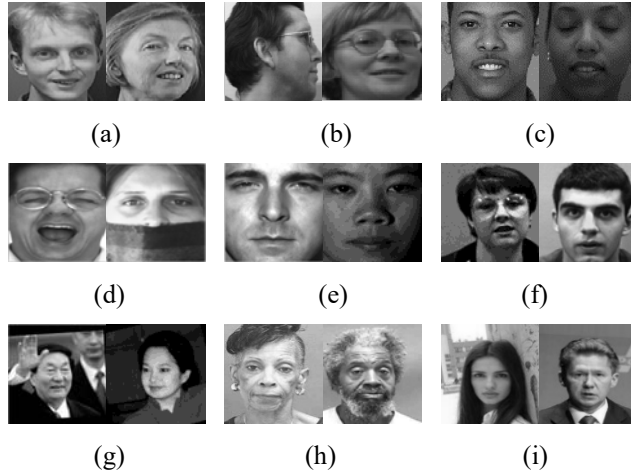


Figure 2 Examples of face images in nine databases: (a) ORL, (b) Georgia Tech, (c) FERET, (d) EYB, (e) AR, (f) Faces94, (g) LFW, (h) MORPH, (i) CelebFaces+.

### 3.2. Whole Comparison of Accuracies and Time

In Table 2, we give the testing accuracies of CorNN and CNN for whole comparison on the nine databases. It is shown that CorNN can perform sometimes as well as CNN (e.g., on ORL and AR), sometimes slightly better (e.g., on Georgia Tech, EYB, LFW, and CelebFaces+), and sometimes slightly worse (e.g., on FERET, Faces94, and MORPH). Hence, CorNN is as good as CNN in terms of accuracy for gender classification.

Table 2 Whole comparison of testing accuracies (%)

Databases	CNN	CorNN
ORL	98.00	98.00
Georgia Tech	97.60	98.22
FERET	94.77	94.29
EYB	98.53	98.69
AR	98.71	98.71
Faces94	96.46	96.18
LFW	87.00	87.14
MORPH	92.73	92.23
CelebFaces+	85.18	85.34

Table 3 Whole comparison of training/testing time (s)

Databases	CNN	CorNN	Reduction
ORL	164.15/2.96	146.65/2.61	10.66%/11.82%
Georgia Tech	166.65/3.33	141.60/2.82	15.03%/15.31%
FERET	501.60/10.03	444.95/9.01	11.29%/10.16%
EYB	1497.95/30.03	1215.80/26.27	18.83%/12.52%
AR	666.90/13.33	577.10/11.79	13.46%/11.55%
Faces94	2068.80/20.82	1695.60/17.37	18.03%/16.57%
LFW	16464/31.31	14334/29.42	12.94%/15.61%
MORPH	50961/54.75	43737/48.32	14.17%/11.74%
CelebFaces+	91848/62.19	79641/54.83	13.26%/11.83%

In Table 3, we give the training/testing time of CNN and CorNN for whole comparison on the nine databases. It is shown that the CorNN takes less than the CNN, with a relative reduction of 11.29%~18.83% training time, and 10.16%~16.57% testing time.

### 3.3. Separate Comparison of Accuracies

In Table 4, we give the testing accuracies of CorNN and CNN for male comparison and female comparison on three databases (i.e., EYB, Faces94 and LFW), with “M” and “F” standing for male and female. It is also shown that CorNN can have as good performance as CNN for male identification and female identification separately, albeit with a slight difference.

Table 4 Separate comparison of testing accuracies (%)

Databases	CNN		CorNN	
	M	F	M	F
EYB	100	97.06	99.65	97.73
Faces94	98.48	94.44	98.94	93.42
LFW	95.00	79.00	95.00	79.28

## 4. Conclusions

Using correlational operation instead of convolutional operation, we propose a CorNN that is equivalent to its corresponding CNN. Moreover, we design an 8-layer CorNN for human face gender classification, together with the learning algorithm CorNN-BP. Additionally, we compare the 8-layer CorNN with the CNN in experiments, verifying that they almost have the same performance in terms of accuracy. However, a CorNN can save a relatively large amount of training time and testing time, due to no requirement of flipping correlational kernels. This is the main advantage of CorNN over CNN.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61175004, the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 20121102110029, and the Project Funding of Post Doctor in Beijing under Grant Q6007011201501.

## References

- [1] Rafi F. H. M., Mitul A. F. An automatic face detection and gender identification from color images using logistic regression. *IEEE Computer Vision and Pattern Recognition*, 2013.
- [2] Golomb B. A., Lawrence D. T., Sejnowski T. J. Sexnet: a neural network identifies sex from human faces. *Advance in Neural Information Processing Systems*, 1991, 572-579.
- [3] Brunellia R., Poggio T. HyperBF networks for gender classification. *DARPA Image Understanding Workshop*, 1995, 311-314.
- [4] Tamura S. H., Kawai H. M. Male/Female identification from 8\*6 very low resolution face images by neural network. *Pattern Recognition*, 1996, 29(2): 331-335.
- [5] Jiao Y., Yang J., Fang Z., et al. Comparing studies of learning methods for human face gender recognition. *Biometric Recognition: the 7th Chinese Conference, CCBR. 2012, Guangzhou, China*, 67-74.
- [6] Verma A., Vig L. Using convolutional neural networks to discover cognitively validated features for gender classification. *2014 International Conference on Soft Computing and Machine Intelligence*, 2014, New Delhi, India, 33-37.
- [7] Hubel D. H., Wiesel T. Receptive fields of signal neurons in the cat’s striate cortex. *Journal of Physiology*, 1962, 160(3): 106-154.



- [8] Fukushima K. Neocognitron: a self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, 1980, 36(4): 193-202.
- [9] Ciresan D. C., Meier U., Gambardella L. M., et al. Convolutional neural network committees for handwritten character classification. 2011 International Conference on Document Analysis and Recognition, Beijing, China, 2011: 1135-1139.
- [10] Khalajzadeh H., Mansouri M., Teshnehlab M. Face recognition using convolutional neural network and simple logistic classifier. *Soft Computing in Industrial Applications*, 2014: 197-207.
- [11] Jialue F., Wei W., Ying W., Yihong G. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 2010, 21(8): 1610-1623.
- [12] Jin J. Q., Fu K., Zhang Ch. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on International Transportation Systems*, 2014, 15(5): 1991-2000.
- [13] Russakovsky O., Deng J., Su H., et al. ImageNet large scale visual recognition challenge, *Journal of Computer Vision*, 2015, 1-42.
- [14] Bengio Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2009, 2(1): 1-127.
- [15] Phillips P. J., Wechsler H., Huang J., Patrick J. Rauss: The FERET database and evaluation procedure for face-recognition algorithms. *Image Vision Computing*. 1998, 16(5): 295-306.
- [16] Kuang-Chih L., Jeffrey H., David K. Acquiring linear subspaces for face recognition under variable lighting. *Pattern Analysis and Machine Intelligence*, 2005, 27(5): 1-15.
- [17] Maetinez A. M. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001, 23(2): 228-233.
- [18] Gary B., Manu R., Tamara B., et al. Labeled faces in the wild: A database for studying face recognition in unconstrained environment. University of Massachusetts, Amherst, Technical Report 07-49, 2007.
- [19] Liu Z., Luo P., Wang X., Tang X. Deep learning face attributes in the Wild. In *Proceedings of the International Conference on Computer Vision*, 2015.