Neurocomputing 275 (2018) 448–461

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A hybrid deep learning CNN–ELM for age and gender classification

Mingxing Duan[a], Kenli Li[b,c,*], Canqun Yang[a], Keqin Li[b,c]

[a] College of Computer Science, National University of Defense Technology, Changsha 410073, China
[b] College of Information Science and Engineering, Hunan University, Changsha, Hunan 410082, China
[c] National Supercomputing Center in Changsha, Changsha, Hunan 410082, China

ARTICLE INFO

Article history:
Received 13 March 2017
Revised 24 August 2017
Accepted 27 August 2017
Available online 8 September 2017

Communicated by Dr. G.-B. Huang

Keywords:
Classification
Convolutional Neural Network
Extreme Learning Machine
Image
Overfitting

ABSTRACT

Automatic age and gender classification has been widely used in a large amount of applications, particularly in human-computer interaction, biometrics, visual surveillance, electronic customer, and commercial applications. In this paper, we introduce a hybrid structure which includes Convolutional Neural Network (CNN) and Extreme Learning Machine (ELM), and integrates the synergy of two classifiers to deal with age and gender classification. The hybrid architecture makes the most of their advantages: CNN is used to extract the features from the input images while ELM classifies the intermediate results. We not only give the detailed deployment of our structure including design of parameters and layers, analysis of the hybrid architecture, and the derivation of back-propagation in this system during the iterations, but also adopt several measures to limit the risk of overfitting. After that, two popular datasets, such as, MORPH-II and Adience Benchmark, are used to verify our hybrid structure. Experimental results show that our hybrid architecture outperforms other studies on the same datasets by exhibiting significant performance improvement in terms of accuracy and efficiency.

1. Introduction

1.1. Motivation

Age and gender classification play a very important role in our social lives, by which we can find whether the persons we contact are “sir” or “madam” and young or old. These behaviors are heavily dependent on our ability to estimate these individual traits: age and gender, which are from facial appearances [1]. These attributes are important in our lives while the ability to estimate them accurately and reliably from facial appearance is still far from satisfying the needs of commercial applications [2].

In order to enhance the ability to estimate or classify these attributes from face images, many methods have been put forward in the past years. Based on cranio-facial changes in feature-position rotation and on skin wrinkle analysis, these attributes have been classified from facial images [3] while a methodology is proposed to classify age and gender automatically from facial images through feature extraction including primary and secondary features [4]. However, these approaches mentioned above have been designed particularly for processing constrained age or gender tasks which are not suitable for practical applications including unconstrained image classification tasks.

The accuracy of age and gender classification depends on two aspects: feature extraction and classification, while feature extraction is a crucial factor for the success of classification. It not only demands the features having the most differentiable characteristics among different classes, but also retains unaltered characteristics within the same class. In recent years, due to its good feature extraction ability, CNN has been highlighted in machine learning and pattern recognition fields. It has achieved state-of-the-art performance in image recognition and can automatically extract the features.

With full consideration of what mentioned above, CNN has been introduced to classify unconstrained age and gender tasks automatically and significant performance has been obtained [2]. More importantly, the unconstrained images are without prior manual filtering, which are as true as real-world applications. CNN has shown great advantages in image recognition while it is the first time to use CNN to process these unconstrained tasks so that we can further improve the accuracy of classification through the fine tuning of its structure or its parameters.

With more discriminative features and more powerful classifier, higher recognition rate will be obtained. In a plain CNN, the full-connection layers are as same as a general single

* Corresponding author at: College of Information Science and Engineering, Hunan University, Changsha, Hunan 410082, China.
E-mail addresses: duanmingxing16@nudt.edu.cn (M. Duan), lkl@hnu.edu.cn (K. Li), canqun@nudt.edu.cn (C. Yang), lik@newpaltz.edu (K. Li).

http://dx.doi.org/10.1016/j.neucom.2017.08.062
0925-2312/

hidden layer feedforward neural network (SLFN) and trained through back-propagation (BP) algorithm. On the one hand, BP algorithm is sensitive to local minima of training errors. On the other hand, SLFN is likely to be over-trained leading to degradation of its generalization performance when it performs BP algorithm [5]. Therefore, the generalization performance of the fully connection layers in the network is probably sub-optimal and they cannot make full use of discriminative features extracted by convolutional layers.

In order to deal with the problems, it is urgent to find a new classifier which owns the similar ability as the full-connection layers or softmax classifier, while it can make full use of the discriminative features. Niu and Suen [6] proposed a hybrid model which integrated the synergy of two superior classifiers including CNN and Support Vector Machine (SVM), and got a better results compared with a plain CNN. In general, the design of SVM is so complicated that is important to find other classifiers with least needing tuning parameters, good classification performance, and high generalization ability to process the same tasks mentioned above. To the best of our knowledge, SVM, Naive Bayes [7], and Extreme Learning Machine (ELM) [8] are three important classification algorithms at present while ELM has been proved to be an efficient and fast classification algorithm because of its good generalization performance, fast training speed, and little human intervene [9]. What's more, ELM and improved ELM, including mixing with other methods, have been widely used to process pattern recognition tasks and obtain a good performance [10].

1.2. Our contributions

In order to make full use of the advantages of CNN and ELM, we propose a hybrid recognition architecture, called CNN–ELM, which is used to process age and gender classification tasks. It not only sufficiently exploits the excellent feature extraction ability of CNN and the outstanding classification property of ELM, but also is used to classify the popular human facial image datasets. At the same time, different effective approaches are adopted to reduce overfitting. With lower time complexity, the hybrid architecture gets a better performance compared with a plain CNN structure which contains the identical convolutional layers. The major contributions of this paper are summarized as follows:

- We propose a new hybrid CNN–ELM method to process age and gender classification aiming at image tasks. It combines Convolutional Neural Networks and Extreme Learning Machine in a hierarchical fashion which is sufficient in applying the advantages of CNN and ELM.
- We present the process of integrating the synergy of hybrid structure in detail, including the design of the layers in CNN, the selection of parameters in hybrid structure, the realization of back-propagation process in this hybrid model, and so on.
- Finally, two popular datasets, such as MORPH-II and Adience Benchmark, are used to verify our hybrid structure. Experiments show that our hybrid structure gets better performance compared with other studies on the same image datasets and also can fulfill the requirements of many real-world application.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 gives preliminary information. Section 4 discusses architecture of our hybrid CNN–ELM model. Section 5 describes merits of hybrid CNN-ELM model. We also analyze the time complexity of hybrid classification in Section 6. The experiments and results are illustrated in Section 7. Finally, we make a conclusion in Section 8.

2. Related work

2.1. Hybrid neural network system

CNN has been successfully applied to various fields, and specially, image recognition is a hot research field. However, few researchers have paid attention on hybrid neural network. Lawrence et al. [11] presented a hybrid neural-network solution for face recognition which made full use of advantages of self-organizing map (SOM) neural network and CNN. That approach showed a higher accuracy compared with other methods used for face recognition at that time. In 2012, Niu and Suen [6] introduced a hybrid classification system for objection recognition by integrating the synergy of CNN and SVM, and experimental results showed that the method improved the classification accuracy. Liu et al. [12] used CNN to extract features while Conditional Random Rield (CRF) was used to classify the deep features. With extensive experiments on different datasets, such as Weizmann horse, Graz-02, MSRC-21, Stanford Background, and PASCAL VOC 2011, the hybrid structure got better segmentation performance compared with other methods on the same datasets. In [13], Xie et al. used a hybrid representation method to process scene recognition and domain adaption. In that method, CNN was used to extract the features meanwhile mid-level local representation (MLR) and convolutional Fisher vector representation (CFV) made the most of local discriminative information in the input images. After that, SVM classifier was used to classify the hybrid representation and achieved better accuracy. Recently, Tang et al. [14] put forward a hybrid structure including Deep Neural Network (DNN) and ELM to detect ship on spaceborne images. In this time, DNN was used to process high-level feature representation and classification while ELM was worked as effective feature pooling and decision making. What is more, extensive experiments were presented to demonstrate that the hybrid structure required least detection time and achieved highter detection accuracy compared with existing relevant methods. Based on the analysis above, we can integrate CNN with other classifiers to improve the classification accuracy. In Sections 4–6, we will present our hybrid CNN-ELM in detail and show its better performance compared with other methods to process the same tasks.

2.2. Age classification

Recently, age and gender classification has received huge attention, which provides direct and quickest way for obtaining implicit and critical social information [15]. Fu et al. [16] made a detailed investigation of age classification and we can learn more information about recent situation from Ref. [2]. Classifying age from the human facial images was first introduced by Kwon et al. [3] and it was presented that calculating ratios and detecting the appearance of wrinkles could classify facial features into different age categorization. After that, the same method was used to model craniofacial growth with a view to both psychophysical evidences and anthropometric evidences [17] while this approach demanded accurate localization of facial features.

Geng et al. [18] proposed a subspace method called AGing pattErn Subspace which was used to estimate age automatically while age manifold learning scheme was presented in [19] to extract face aging features and a locally adjusted robust regressor was designed to predict human ages. Although these methods have shown many advantages, the requirement that input images need to be near-frontal and well-aligned is their weakness. It is not difficult to find that the datasets in their experiments are constrained, so that these approaches are not suited for many practical applications including unconstrained image tasks.

Last year, many methods have been proposed to classify age and gender. Chang and Chen [20] introduced a cost-sensitive ordinal hyperplanes ranking method to estimate human age from facial images while a novel multistage learning system which is called grouping estimation fusion (DEF) was proposed to classify human age. Li et al. [21] estimated age using a novel feature selection method and shown advantage of the proposed algorithm from the experiments. Although these method mentioned above have shown lots of advantages, they are still relied on constrained images datasets, such as FG-NET [22], MORPH [23], FACES [24].

All of these methods mentioned above have been verified effectively on constrained datasets for age classification which are not suitable for unconstrain images in practical applications. Our proposed method not only automatically classifies age and gender from face images, but also deals with the unconstrain face image tasks effectively.

2.3. Gender classification

Although more and more researchers have found that gender classification has played an important role in our daily life, few learning-based machine vision approaches have been put forward. Makinen and Raisamo [25] made a detailed investigation of gender classification while we can learn more about its recent trend from Ref. [2]. In the following, we briefly review and summarize relevant methods.

Golomb et al. [26] were some of the early researchers who used a neural network which was trained on a small set of near-frontal facial image dataset to classify gender. Moghaddam and Yang [27] used SVM to classify gender from facial images while Baluja and Rowley [28] adopted AdaBoost to identify human gender from facial images. After that, Toews and Arbel [29] presented a viewpoint-invariant appearance model of local scale-invariant features to classify age and gender.

Recently, Yu et al. [30] put forward a study and analysis of gender classification based on human gait while revisiting linear discriminant techniques was used to classify gender [31]. In [1], Eidinger et al. not only presented new and extensive dataset and benchmarks to study age and gender classification, but also designed a classification pipeline to make full use of what little data was available. In [9], a semantic pyramid for gender and action recognition was proposed by Khan et al. and the method is fully automatic while it does not demand any annotations for a person upper body and face. Chen et al. [32] used first names as facial attributes and modeled the relationship between first names and faces. They used the relationship to classify gender and got higher accuracy compared with other methods. Last year, Han et al. [33] used a generic structure to estimate age, gender, and race.

Although most of the approaches mentioned above make lots of progress for age classification, they are aimed at either constrain imaging condition or non-automated classification methods. Our hybrid CNN–ELM structure is not only suitable to process unconstrain face images, but also able to automatically classify age and gender tasks based on facial images.

3. Preliminary information

3.1. Deep Convolutional Neural Networks

Convolutional Neural Network [34], which usually includes input layer, multi-hidden layers, and output layer, is a deep supervised learning architecture and often made up of two parts: an automatic feature extractor and a trainable classifier. CNN has shown remarkable performance on visual recognition [35]. When we use CNNs to process visual tasks, they first extract local features from the input images. In order to obtain higher order features, the subsequent layers of CNNs will then combine these features. After that, these feature maps are finally encoded into 1-D vectors and a trainable classifier will deal with these vectors. Because of considering size, slant, and position variations for images, feature extraction is a key step during classification of images. Therefore, with the purpose of ensuring some degree of shift, scale, and distortion invariance, CNNs offer local receptive fields, shared weights, and downsampling. Fig. 1 is a basic architecture of CNNs.

It can be seen from Fig. 1 that CNNs mainly include three parts: convolution layers, subsampling layers and classification layer. The main purpose of convolutional layers is to extract local patterns and the convolutional operations can enhance the original signal and lower the noise. Moreover, the weights of each filtering kernels in each feature maps are shared, which not only reduce the free parameters of networks, but also lower the complication of relevant layers. The outputs of the convolutional operations contain several feature maps and each neuron in entire feature maps connects the local region of the front layers. Subsampling is similar to a fuzzy filter which is primary to re-extract features from the convolutional layers. With the local correlation principle, the operations of subsampling not only eliminate non-maximal values and reduce computations for previous layer, but also improve the ability of distortion tolerance of the networks and provide additional robustness to position. These features will be encoded into a 1-D vectors in the full connection layer. After that, these vectors will be categorized by a trainable classifier. Finally, the whole neural network will be trained by a standard error back propagation algorithm with stochastic gradient descent [36]. The purpose of training CNNs is to adjust the entire parameters of the system, i.e., the weights and biases of the convolution kernel, and we will use the fine-tuned CNNs to predict the classes, such as label, age, and so on, from an unknown input image datasets.

3.2. Extreme machine learning model

ELM was first proposed by Huang et al. [8,10,37] which was used for the single-hidden-layer feedforward neural networks (SLFNs). The input weights and hidden layer biases are randomly assigned at first, and then the training datasets to determine the

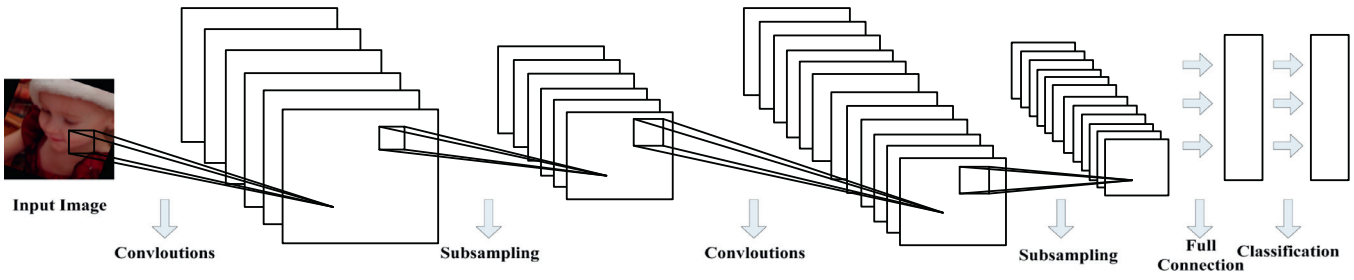


Fig. 1. Structure of CNN for visual recognition.

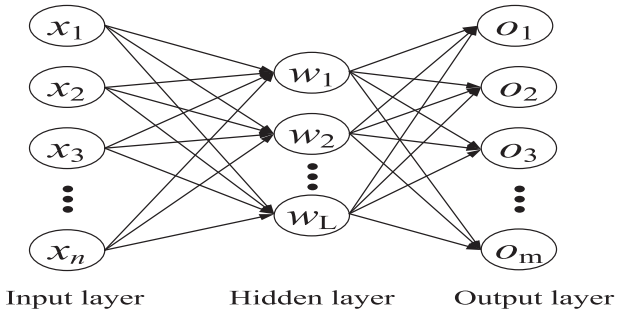


Fig. 2. A basic structure of ELM.

output weights of SLFNs are combined. Fig. 2 is a basic structure of ELM. For N arbitrary distinct samples (x_i, t_i), $i = 1, 2, ..., N$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$, $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T$. Therefore, the ELM model can be written as

$$\sum_{j=1}^{L} \beta_j g_j(\mathbf{x}_i) = \sum_{j=1}^{L} \beta_j g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j) = \mathbf{o}_i \quad (i = 1, 2, \ldots, N), \tag{1}$$

where $\beta_j = [\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}]^T$ expresses the jth hidden node weight vector while the weight vector between the jth hidden node and the output layer can be described as $\mathbf{w}_j = [w_{1j}, w_{2j}, \ldots, w_{nj}]^T$. The threshold of the jth hidden node can be written as b_j and $\mathbf{o}_i = [o_{i1}, o_{i2}, \ldots, o_{im}]^T$ denotes the ith output vector of ELM.

We can approximate the output of ELM if activation function $g(x)$ with zero error which means as Eq. (2):

$$\sum_{i=1}^{N} ||\mathbf{o}_i - \mathbf{t}_i|| = 0. \tag{2}$$

Therefore, Eq. (1) can be described as Eq. (3):

$$\sum_{j=1}^{L} \beta_j g_j(\mathbf{x}_i) = \sum_{j=1}^{L} \beta_j g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j) = \mathbf{t}_i \quad (i = 1, 2, \ldots, N). \tag{3}$$

Finally, Eq. (3) can be simply expressed as Eq. (4):

$$\mathbf{H}\beta = \mathbf{T}, \tag{4}$$

where $\mathbf{H}$ expresses the hidden layer output matrix, and $\mathbf{H} = \mathbf{H}(\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_L, b_1, b_2, ..., b_L, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$. Therefore, $\mathbf{H}$, β, and $\mathbf{T}$ can be written as follows:

$$[h_{ij}] = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \ldots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \ldots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}, \tag{5}$$

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \ldots & \beta_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{L1} & \beta_{L2} & \ldots & \beta_{Lm} \end{bmatrix}, \tag{6}$$

and

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & \ldots & t_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N1} & t_{N2} & \ldots & t_{Nm} \end{bmatrix}. \tag{7}$$

After that, the smallest norm least-squares solution of Eq. (4) is

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{8}$$

where $\mathbf{H}^{\dagger}$ denotes the Moore–Penrose generalized the inverse of matrix $\mathbf{H}$. The output of ELM can be expressed as Eq. (9):

$$f(\mathbf{x}) = h(\mathbf{x})\beta = h(\mathbf{x})\mathbf{H}^{\dagger}\mathbf{T}. \tag{9}$$

From the description above, the process of ELM can be described as follows. At the beginning, ELM was randomly assigned the input weights and the hidden layer biases $(\mathbf{w}_i, b_i)$. After that, we calculate the hidden layer output matrix $\mathbf{H}$ according to Eq. (5). Then, by using Eq. (8), we can obtain the output weight vector β. Finally, we can classify the new dataset according to the above training process.

ELM is not only widely used to process binary classification [38–41], but also used for multi-classification due to its good properties. As we have mentioned in part 3.1, CNNs show excellent performance on extracting feature from the input images, which can reflect the important character attributes of the input images. Therefore, we can integrate the advantages of CNNs and ELM based on the analysis above, which means CNNs extract features from the input images while ELM classify the input feature vectors.

4. Architecture of our hybrid CNN–ELM model

In this section, we present the design of our hybrid structure in detail. Fig. 3 is the architecture of our CNN–ELM. It can be seen from the figure that our network includes two stages, feature extraction and classification. The stage of feature extraction contains the convolutional layer, contrast normalization layer, and max pooling layer. We also detailedly give the correlative parameters, such as, the number of each filters, the size of each feature maps, the kernel size of each filters, and the stride of each sliding windows. For example, the first convolutional layer consists of 96 filters, and the size of its feature map is 56×56 while its kernel size is 7 and the stride of the sliding window is 4. A single convolution layer is implemented after the two stages, and a full connection layer converts the feature maps into 1-D vectors which is beneficial to the classification. Finally, we combine the ELM structure with our designed CNN model, and we will use this hybrid model to classify the age and gender tasks. We will detailedly present the design of each part of hybrid structure in following sections.

4.1. Design of our hybrid structure

4.1.1. Convolutional layer

In the convolutional layer, convolutions which are performed between the previous layer and a series of filters, extract features from the input feature maps [42,43]. After that, the outputs of the convolutions will add an additive bias and an element-wise nonlinear activation function is applied on the front results. We use the ReLU function as the nonlinear function in our experiment. In general, η_{ij}^{mn} denotes the value of an unit at position (m, n) in the jth feature map in the ith layer and it can be expressed as Eq. (10):

$$\eta_{ij}^{mn} = \sigma\left(b_{ij} + \sum_{\delta} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ij\,\delta}^{pq} \eta_{(i-1)\delta}^{(m+p)(n+q)}\right), \tag{10}$$

where b_{ij} represents the bias of this feature map while δ indexes over the set of the feature maps in the $(i-1)$th layer which are connected to this convolutional layer. $w_{ij\delta}^{pq}$ denotes the value at the position (p, q) of the kernel which is connected to the kth feature map and the height and width of the filter kernel are P_i and Q_i.

The convolutional layer offers a nonlinear mapping from the low level representation of the images to the high level semantic understanding. In order to be convenient to later computations, Eq. (10) can be simply denoted as follows:

$$\eta_j = \sigma\left(\sum w_{ij} \bigotimes \eta_{(i-1)}\right), \tag{11}$$

where $\otimes$ expresses the convolutional operation while w_{ij}, which will be randomly initialized at first and then trained with **BP** neural network [44,45], denotes the value of the ith layer in the jth

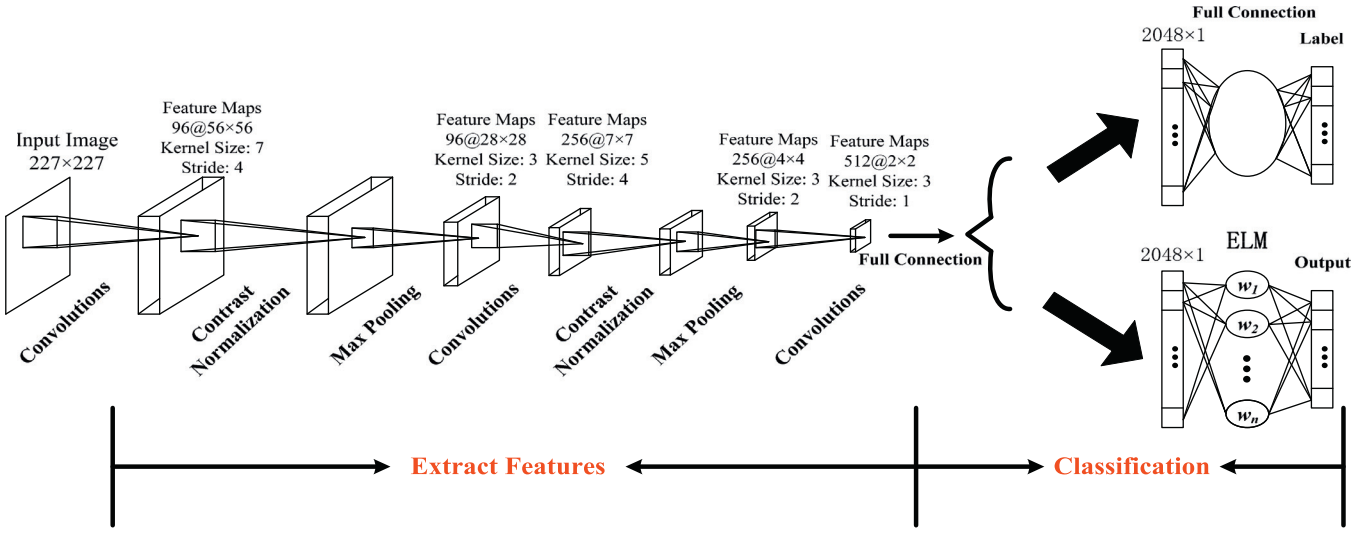


Fig. 3. Full schematic diagram of our network architecture.

feature map. $\eta_{(i-1)}$ is the outputs of the $(i-1)$ layer and η_j is defined as the outputs of the jth feature map in the convolutional layer. Different sizes of the input feature maps have various effects on the accuracy of classification. Large size of a feature map means good features learned by the convolutional operations with the high cost of the computations while small size reduces the computation cost degrading the accuracy of the classification. Making a comprehensive consideration of the factors mentioned above and by lots of experiments, we set the size of the input feature map as 227×227 which is showed in Fig. 3.

4.1.2. Contrast normalization layer

The goal of the local contrast normalization layer is not only to enhance the local competitions between one neuron and its neighbors, but also to force features of different feature maps in the same spatial location to be computed, which is motivated by the computational neuroscience [45,46]. In order to achieve the target, two normalization operations, i.e., subtractive and divisive, are performed. In this time, η_{mnk} denotes the value of an unit at position (m, n) in the kth feature map. We have

$$z_{mnk} = \eta_{mnk} - \sum_{p=-\frac{P_i-1}{2}}^{\frac{P_i-1}{2}} \sum_{q=-\frac{Q_i-1}{2}}^{\frac{Q_i-1}{2}} \sum_{j=1}^{J_i} \varepsilon_{pq} \eta_{(m+p)(n+q)j}, \tag{12}$$

where ε_{pq} is a normalized Gaussian filter with the size of 7×7 at the first stage and 5×5 at the second stage. z_{mnk} not only represents the input of the divisive normalization operations, but also denotes the output of the subtractive normalization operations. Eq. (13) expresses the operator of the divisive normalization:

$$\eta_{mnk} = \frac{z_{mnk}}{\max\left(M, M(m,n)\right)}, \tag{13}$$

where

$$M(m,\ n) = \sqrt{\sum_{p=-\frac{P_i-1}{2}}^{\frac{P_i-1}{2}} \sum_{q=-\frac{Q_i-1}{2}}^{\frac{Q_i-1}{2}} \sum_{j=1}^{J_i} \varepsilon_{pq} \eta^2_{(m+p)(n+q)j}}, \tag{14}$$

and

$$M = \left(\sum_{m=1}^{s1} \sum_{n=1}^{s2} M(m,\ n) \right) / (s1 \times s2). \tag{15}$$

During the whole contrast normalization operations above, the Gaussian filter ε_{pq} is calculated with the zero-padded edges, which means that the size of the output of the contrast normalization operations is as same as its input.

4.1.3. Max pooling layer

Generally speaking, the purpose of pooling strategy is to transform the joint feature representation into a novel, more useful one which keeps crucial information while discards irrelevant details. Each feature map in the subsampling layer is getting by max pooling operations which are carried out on the corresponding feature map in convolutional layers. Eq. (16) is the value of a unit at position (m, n) in the jth feature map in the ith layer or subsampling layer after max pooling operation:

$$\eta_{ij}^{mn} = \max\left\{ \eta_{(i-1)j}^{mn},\ \eta_{(i-1)j}^{(m+1)(n+1)},\ \ldots,\ \eta_{(i-1)j}^{(m+P_i)(n+Q_i)} \right\}. \tag{16}$$

The max pooling operation generates position invariance over larger local regions and downsamples the input feature maps. In this time, the numbers of feature maps in the subsampling layer are 96 while the size of the filter is 3 and the stride of the sliding window is 2. The aim of max pooling action is to detect the maximum response of the generated feature maps while reduces the resolution of the feature map. Moreover, the pooling operation also offers built-in invariance to small shifts and distortions. The procedures of other convolutional layers and subsampling layers which we have not told are as same as the layers mentioned above, except with a different kernel size or stride.

4.1.4. ELM classification layer

After the convolution and subsampling operations, ELM is used to classify the 1-D vectors which are converted from feature maps. As we have mentioned in part 3.2, it only updates the output weights while input weights and hidden-layer biases are randomly set, thus we will randomly generate the input parameters and calculate the output weights during the training stage. The whole process without iteration operation improves the neural network generalization ability. From Fig. 3, we can find that the output (containing 2048×1 dimensionality) of full-connection layer is the input of ELM while the numbers of hidden nodes are variables which will be shown in our experiments.

The connection between ELM and convolutional network is also a critical process and we can see from Fig. 3 that our input of ELM is the output of the full connection layer whose preceding layer is a convolutional layer. Forward-propagation and back-propagation operations are the principal parts in our hybrid architecture and we analyze them in detail in following sections.

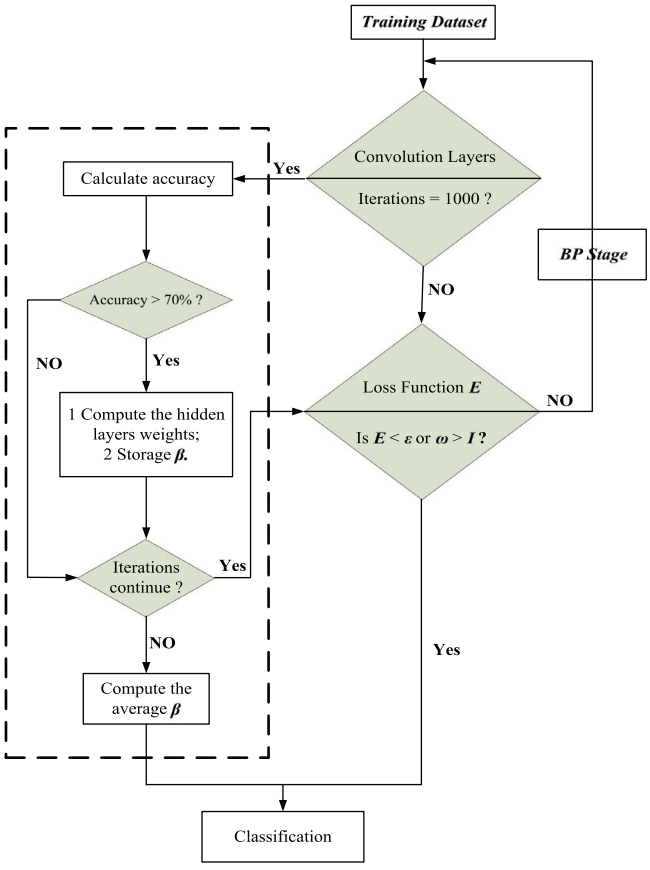


Fig. 4. The simple process of our hybrid structure.

4.2. Process of our CNN–ELM

There is no double that our hybrid structure needs to tune the parameters of CNN from the learning process during the training stage at first while ELM has not been invoked. After that, for every 1000 iterations, we will verify the accuracy of the structure, i.e., whether it has fine-tuned the parameters and extracted discriminative features. If the accuracy nearly reaches 70%, the ELM layer will be invoked. At that time, we will first compute the hidden layer weights, and cache the intermediate β matrices, then the hybrid structure will be used to verify its accuracy. When the training accuracy of our hybrid structure gets nearly 100% or the whole iterations exceed the setting *max* iterations, we stop the training process and calculate the average of intermediate β matrices. Finally, our hybrid CNN–ELM will be used for classifying age and gender tasks during test stage. The steps are summarized as follows:

Step 1: Tune the parameters of CNN during the training stage when the connection between convolutional layers and output labels is full connection layers.
Step 2: Compute the hidden layer weights and cache the intermediate β matrices, meanwhile verify the accuracy of fine-tuned network.
Step 3: Stop the training process and calculate the average of $\boldsymbol{\beta}$.
Step 4: Classify the unknown dataset using our hybrid structure.

Fig. 4 presents a simple process of our hybrid structure and E denotes the Loss Function while ω expresses the whole regulation iterations. During the training stage, enough experiments have been carried out. We will have a test for our hybrid structure every 1000 iterations, and hidden node weight vectors of ELM will be computed according to Eq. (8) at that time. Finally, the hidden node weight vectors $\boldsymbol{\beta}$ will be the average of different hidden node weight vectors during the test stage.

In order to fine tune the network, we have trained our structure for more than 10K iterations. This process is main to tune the parameters of CNN and makes it own the ability of extracting discriminative features. During the training stage, we will obtain $\hat{\beta}$ according to Eq. (8), which is prepared to classify the unknown dataset. We will test our hybrid structure under different hidden nodes. In the following, we will present the implementation of our hybrid CNN–ELM.

4.2.1. Training stage using hybrid structure

Fig. 4 shows that the training stage not only tunes the parameters of convolutional layer, but also achieves the corresponding hidden layer weights of ELM. The feed-forward process of our hybrid structure is as same as a plain CNN while every 1000 iterations, ELM layers, instead of full connection layers, will be invoked and corresponding hidden layer weights will be calculated. At the same time, intermediate results β matrices will be stored in the memory using for final average results. Algorithm 1 presents the approximate process in the training stage when the accuracy arrives 70%.

Algorithm 1 Feed-forward process.

Input:
Training samples $\chi = \{(\mathbf{x}_i, \mathbf{t}_i) \mid \mathbf{x}_i \in \mathbf{R}^n \times \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, 2, \ldots, N\}$;
Convolutional Net.layers = S;
Maximum iterations: I;
Maximum precision: ε;
Numbers of iteration: ω.
Output:
Obtain the error e and LossFunction E.

```
Parse the training samples;
for l from 2 to S
    if net.layer[l] equals to Convolutation layer
        Randomly generate the weights W_ij^pq and bias b_ij;
        Extract features according to (Eq. 10) or (Eq. 11);
        Compute the outputs of contrast normalization layer according to (Eq. 13);
    else if net.layer[l] equals to Max Pooling layer
        Compute the feature maps according to (Eq. 16);
    else if net.layer[l] equals to Full Connnection layer
        Transform the 2-D feature maps of the last convolution layer into 1-D vectors (x_i^l ∈ R^n', i = 1, 2, ..., N);
    end if
end for
Randomly generate the input weights and bias of ELM;
Compute the hidden layer output matrix H according to (Eq. 1);
Obtain the output weight vectors β according to (Eq. 8);
Compute the output of hybrid CNN-ELM, y = Hβ;
Cache β in the memory;
Compute the error e = 1/2 Σ_{k=1}^{m} (t_i(k) − o_i(k))^2;
Compute the LossFunction E = 1/(2N) Σ_{i=1}^{N} Σ_{k=1}^{m} (t_i(k) − o_i(k))^2;
ω = ω + 1;
if E < ε or ω > I
    Compute the average of β;
    Wait for classification stage;
else Call Algorithm 2.
end if
```

As we have mentioned in part 4.2, when ELM classifier works and the whole iterations continue, the system will adopt stochastic gradient descent to tune the relevant parameters of the entire convolutional networks. During process of back propagation, the operations between convolutional layer and subsampling layer or subampling layer and convolutional layer are as same as a single convolution neural network. Note that ELM is just a feed forward algorithm, so we just transform the the error from ELM' output layer to the convolutional layers while we do not tune the parameters of ELM during the process. How to calculate the gradients of ELM and transform them to convolutional part is a key step.

Based on the analysis above, we can give a detail algebraic relation for our hybrid CNN–ELM model. $O_o(k)$ expresses the output of kth sample while its ideal output is $T_o(k)$. The weights between the output layer and hidden layer can be written as β_{ho} and $H_{ho}(k)$ denotes the output of hidden layer. $x_i(k)$ is the input of kth sample and w_{ih} signifies the weight of input layer. $H_{ih}(k)$ denotes the input of kth sample in hidden layer and e is error. We use $\delta_s^l(k)$ to denote the kth sample local gradient which is the sth feature map in the lth layer. Therefore, we can obtain the equations as follows:

$$e = \frac{1}{2}\sum_{o}^{m}(T_o(k) - O_o(k))^2. \tag{17}$$

According to Eq. (17), we can know that e is a multivariate function which is about

$$O_o(k) = \sum_{h}^{L} H_{ho}(k)\beta_{ho}, \tag{18}$$

$$H_{ho}(k) = g(H_{ih}(k)), \tag{19}$$

$$H_{ih}(k) = \sum_{i}^{m} w_{ih}x_i(k) + b_h. \tag{20}$$

Therefore, according to BP theory, we can know:

$$\begin{aligned}\frac{\partial e}{\partial \beta_{ho}} &= \frac{\partial \frac{1}{2}\sum_{o}^{m}(T_o(k) - O_o(k))^2}{\partial \beta_{ho}} \\ &= \frac{\partial \frac{1}{2}\sum_{O}^{m}(T_o(k) - \sum_{h}^{L} H_{ho}(k)\beta_{ho})^2}{\partial \beta_{ho}} \\ &= (T_o(k) - O_o(k))(-H_{ho}(k)) \\ &= \delta_o(k)(-H_{ho}(k)),\end{aligned} \tag{21}$$

while

$$\begin{aligned}\frac{\partial e}{\partial w_{ih}} &= \frac{\partial e}{\partial H_{ho}(k)}\frac{\partial H_{ho}(k)}{\partial H_{ih}(k)}\frac{\partial H_{ih}(k)}{\partial w_{ih}} \\ &= \frac{\partial \frac{1}{2}\sum_{O}^{m}(T_o(k) - \sum_{h}^{L} H_{ho}(k)\beta_{ho})^2}{\partial H_{ho}(k)}\frac{\partial g(H_{ho}(k))}{\partial H_{ih}(k)} \\ &\quad \times \frac{\partial \sum_{i}^{m}(w_{ih}x_i(h) + b_h)}{\partial w_{ih}} \\ &= -\left(\sum_{o}^{m}(T_o(k) - O_o(k))\beta_{ho}\right)g'(H_{ih}(k))x_i(k) \\ &= \delta_h(k)x_i(k) \\ &= -\left(\sum_{o}^{m}\delta_o\beta_{ho}\right)g'(H_{ih}(k))x_i(k).\end{aligned} \tag{22}$$

Therefore, we can obtain the relation between $\delta_h(k)$ and $\delta_o(k)$ as follows:

$$\delta_h = -\left(\sum_{o}^{m}\delta_o\beta_{ho}\right)g'(H_{ih}(k)). \tag{23}$$

Finally

$$\begin{aligned}\frac{\partial e}{\partial b_h} &= \frac{\partial e}{\partial H_{ho}(k)}\frac{\partial H_{ho}(k)}{\partial H_{ih}(k)}\frac{\partial H_{ih}(k)}{\partial b_h} \\ &= \frac{\partial \frac{1}{2}\sum_{O}^{m}(T_o(k) - \sum_{h}^{L} H_{ho}(k)\beta_{ho})^2}{\partial H_{ho}(k)}\frac{\partial g(H_{ho}(k))}{\partial H_{ih}(k)} \\ &\quad \times \frac{\partial \sum_{i}^{m}(w_{ih}x_i(h) + b_h)}{\partial b_h} \\ &= -\left(\sum_{o}^{m}(T_o(k) - O_o(k))\beta_{ho}\right)g'(H_{ih}(k)) \\ &= -\left(\sum_{o}^{m}(T_o(k) - O_o(k))\beta_{ho}\right)g'(H_{ih}(k)) \\ &= \delta_h(k) \\ &= -\sum_{o}^{m}\delta_o\beta_{ho}g'(H_{ih}(k)).\end{aligned} \tag{24}$$

After that, we will compute the local gradient in the full connection layer. Compared with a plain CNN, our hybrid architecture also transforms the feature maps into 1-D vectors in the process of forward propagation, so we just need to transform the local gradient in the input layer of ELM to convolutional layer. Therefore, the whole process of back propagation can be written as Algorithm 2.

Algorithm 2 Back-propagation algorithm.

Input:
Real output $O_o(k)$, k = 1, 2, ..., N, o = 1, 2, ..., m;
Ideal output $T_o(k)$, k = 1, 2, ..., N, o = 1, 2, ..., m;
Convolutional Net.layers = S;
Weights in different layers;
Biases in different layers;
Inputs y_i^l;
Outputs y_o^l.

Output:
Updated weights and biases in each layers.

Compute the error according to (Eq. 17);
Compute the local gradient in the output layer according to (Eq. 21);
Compute the local gradient in the hidden layer according to (Eq. 22);
for l from S to 2
  if *net.layer*[l] equals to *Max Pooling* layer
    Compute the local gradient δ;
    Compute the modified weights coefficient $\Delta w = \eta\delta y_o^{(l-1)}$;
    Compute the modified biases coefficient $\Delta b = \eta\delta$;
    Update the whole weights in this layer $w' = w + \Delta w$;
    Update the whole biases in this layer $b' = b + \Delta b$;
  else if *net.layer*[l] equals to *Convolution* layer
    Expand the size of matrix $\delta^{(l-1)}$ to equal to the size of lth feature maps;
    Compute the local gradient δ;
    Compute the modified weights coefficient $\Delta w = \eta\delta y_o^{(l-1)}$;
    Compute the modified biases coefficient $\Delta b = \eta\delta$;
    Update the whole weights in this layer $w' = w + \Delta w$;
    Update the whole biases in this layer $b' = b + \Delta b$.
  end if
end for

4.2.2. Classification process

When we have fine tuned our structure and verify its accuracy meeting our setting standard, we will classify the unknown subjects into different age or gender categories. The information is

extracted from input dataset to hidden layers, and then classified as corresponding output. The steps are as follows:

Step 1: Extract the features with convolutional layers from the unknown subjects.
Step 2: Classify the features using our fine-tuning structure.

During the experiments, although our structure gains higher accuracy compared with other algorithms in terms of the same problems, we find that misclassification is still existing mainly due to the challengeable Adience Benchmark. Meanwhile, we find our structure needs more memory because of caching the hidden layer weights and calculating Moore–Penrose generalized inverse matrix $\mathbf{H}^{\dagger}$. Note that our hybrid classifier only updates the output weights while the input weights and biases of hidden layer and weights are randomly generated, which not only improves the learning speed, but also limits the risk of overfitting. Based on our experimental results, improving the numbers of hidden layer nodes of our structure can improve the classification accuracy, but if numbers of these exceed a specific scope (nearly 4500 in our experments), the accuracy will be degraded mainly due to more commutation cost, more memory, information losing, and aggravating overfitting. In all, our structure not only accelerates the learning speed, but also improves the classification accuracy.

5. Merits of our hybrid structure

Our expectation is that our hybrid model will outperform other individual classifier, which means that our structure can be able to compensate the limit of the classification ability of CNN and make full use of the advantage of ELM. During the training stage, we mainly tune the convolutional networks and we have a test every 1000 iterations. When the accuracy reaches 70%, the hybrid system will adjust the parameters of ELM. In order to gain good generalization performance, we obtain the average hidden node weights of ELM in our hybrid structure. This process not only exploits the good ability of feature extraction in convolutional network, but also makes the most of the advantage of our ELM structure including good generalization performance, fast training speed, and little human intervene, which accelerates the whole learning speed. Meanwhile, several measures including data augmentation, different dropouts, and so on, are used to reduce the risk of overfitting. Experiments verify that our hybrid structure has realized the expectation.

6. Time complexity of hybrid classifiers

In this section, we analyze the complexity of our hybrid structure. The time complexity of all convolutional layers can be written as follows:

$$O\left(\sum_{l=1}^{d}\sigma_{l-1}\cdot\alpha_l^2\cdot\sigma_l\cdot\beta_l^2\right), \tag{25}$$

where l denotes the index of a convolutional layer while d is its depth. σ_l expresses the number of filters in the lth layer and the spatial size of a filter can be written as α_l. σ_{l-1} is the number of input channels in the lth layer while β_l denotes the spatial size of the output feature maps.

As mentioned in [47], the training stage has occupied most time during the experiments while the process of fine-tuning of parameters in the convolutional layers costs most training time. We have not considered the full connection layer and max pooling layers in equation above, because these layers cost 10% computational time approximately. Note that Eq. (25) is not the real running time, due to the system sensitivity to experimental environment. After ten times experiments, we find that the whole process costs less time than a plain CNN, which can be approximately seen that the time complexity of our hybrid structure is much lower compared with other algorithms. Furthermore, our hybrid structure can increase the performance of face recognition and classification.

Table 1
The dataset using for age and gender classification.

Total number of photos	26,580
Total number of subjects	2284
Number of age groups or labels	8
Gender labels	yes
In the wild	yes
Subject labels	yes

7. Experiments

Our method is implemented using the publicly available code of *cuda-convnet* [48] and Caffe [49]. The whole networks in this paper are trained on a single GeForce GTX 750. At first, we will reshape the input image as 256×256 pixels, and then a 224×224 crop will be selected from the center or the four corners from the entire processed image above. We also adopt different dropout measures to limit the risk of overfitting. Training each network needs nearly ten hours while classifying a single image about age or gender nearly costs 600ms. Each experiment has been conducted ten times and we achieve its corresponding average.

7.1. Adience benchmark

In this work, we use the recently released Adience benchmark [1,2], which is designed for age and gender classification, to test our hybrid structure. To this end, the benchmark of face photos is made of images and they are from smart-phone devices. Due to these images uploaded to Flickr without prior manual filtering, these images are highly unconstrained, which are as true as the challenges of real-world applications. Therefore, the images include all variations in appearance, noise, pose, lighting and more, which mean that the photos are taken without careful preparation or posing. The datasets are obtained from the Computer Vision Lab at the Open University of Israel (OUI) [50] and showed in Table 1. Meanwhile, Table 2 shows the different age categories of the Adience benchmark detailedly.

7.2. MORPH-II database

MORPH-II [23] has approximately 55,000 facial images, in which 46,645 of the images are Male, 8487 are Female. The database is used to verify our CNN–ELM performance.

7.3. Age classification with adience benchmark

7.3.1. Error rate under different conditions

We test our hybrid system on the Adience benchmark, and we compare our algorithm with a plain CNN which includes the identical convolutional layers. We train our hybrid CNN–ELM model using mini-batch stochastic gradient descent with 0.7. During the fine-tuning of parameters in our hybrid structure, the learning rate at beginning is designed as 10^{-3} while it decreases to 10^{-4} after 10K iterations. When it arrives 12K iterations, the rate is set as 10^{-5}. Fig. 10(a) shows the error rate during the training stage while the hidden nodes are 3000. FC stage means that the output layer just uses full connection layers while ELM stage shows that ELM classifier works at that time.

From Fig. 5(a), we can find that the training error rates of the two algorithms gets high at the beginning while they nearly tend to zero. They can fine tune parameters automatically through

Table 2
The different age categories of the Adience benchmark.

	0–2	4–6	8–13	15–20	25–32	38–43	48–53	60–	Total
Male	745	928	934	734	2308	1294	392	442	8192
Female	682	1234	1360	919	2589	1056	433	427	9411
Both	1427	2162	2294	1653	4897	2350	825	869	19,487

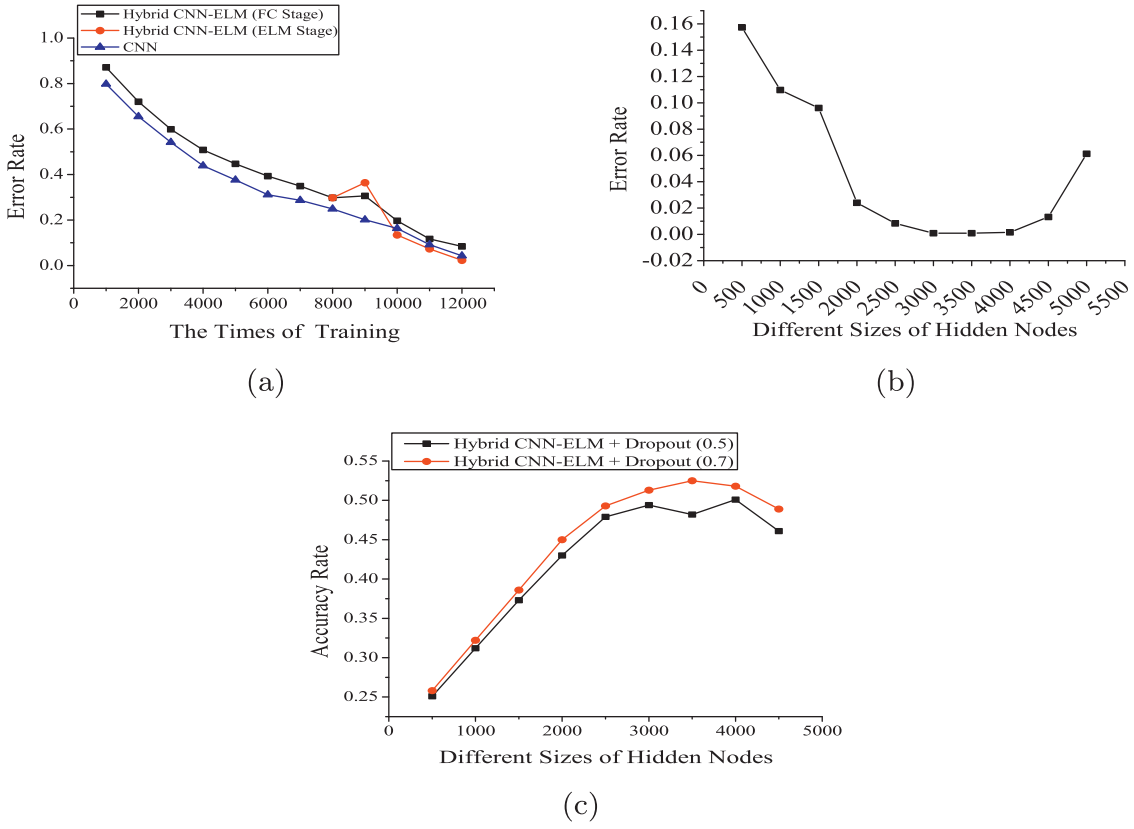


Fig. 5. Experimental results of age classification on Adience Benchmark. (a) Error rate under different training times; (b) error rate under different sizes of hidden nodes; (c) the accuracy of age classification under different hidden layer nodes.

learning process and get better results. At the same time, we also find that their speed of learning is faster during 10K iterations than the speed in the end of training stage. When the error rate approaches to 30%, our ELM classifier will work. At that time, we find that the error rate rebounds at first, and then sharply approaches to 0. Finally, the error rate increases because we use the average of $\boldsymbol{\beta}$ matrix for the hidden layer weights of ELM classifier. At the same time, the training error rate of full connection (FC) layer has a fluctuation due to the transformation of different classifiers. During the whole training stage, the error rates of FC stage are higher than a plain CNN due to its output using a softmax classifier while the error rate in ELM stage gets lower quickly compared with a plain CNN because of ELM' good learning ability and good classification performance. Based on the analysis above, we can conclude that our ELM classifier has fast training speed and can get better training results. In all, our hybrid structure can not only quickly tune the parameters automatically, which makes sure that the convolutional layers extract the discriminative features in favour of classification, but also achieve better parameters for classifiers.

We also show that the training error rate under different sizes of hidden nodes in Fig. 5(b) and the number of iterations is 12K. It can be seen from the figure that with the increase of hidden layer nodes, the error rate declines significantly at the first while it stays smoothly when the hidden nodes are between 2500 and 4000. However, when the hidden nodes tend to 4500, the error rate rises slowly. The reason for that is that continuously increasing the hidden nodes aggravates overfitting, which causes the degradation of the performance of classification. Therefore, we will set the hidden node as 3500 in the next experiments.

Table 3
The accuracy of age classification.

Method	Accuracy
LBP [51]	40.7% ± 2.0%
LBP + FPLBP [52]	44.1% ± 2.4%
LBP + FPLBP + Dropout 0.5 [1]	44.9% ± 2.2%
LBP + FPLBP + Dropout 0.8 [1]	45.2% ± 2.6%
Best from [2]	50.7% ± 5.1%
Proposed CNN-ELM + Dropout 0.5	51.4 % ± 5.2%
Proposed CNN-ELM + Dropout 0.7	52.3% ± 5.7%

7.3.2. Accuracy of age classification

We present our results for age classification while we also mix our structure with dropout layer between convolutional layer and classification layer. It is no doubt that the dropout structure can limit the risk of overfitting. We set two different dropout ratios as 0.5, 0.7, respectively (50% or 70% probability to set the output value of a neural as 0). Each experiment has been done more than ten times, then we obtain their corresponding averages. Therefore, our accuracy includes mean accuracy ± standard deviations.

From Table 3, we find that our hybrid structure can get the highest accuracy among the compared algorithms. Note that the

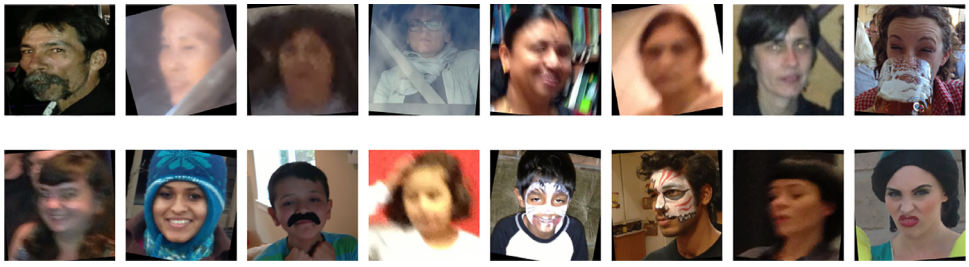


Fig. 6. Age misclassification.

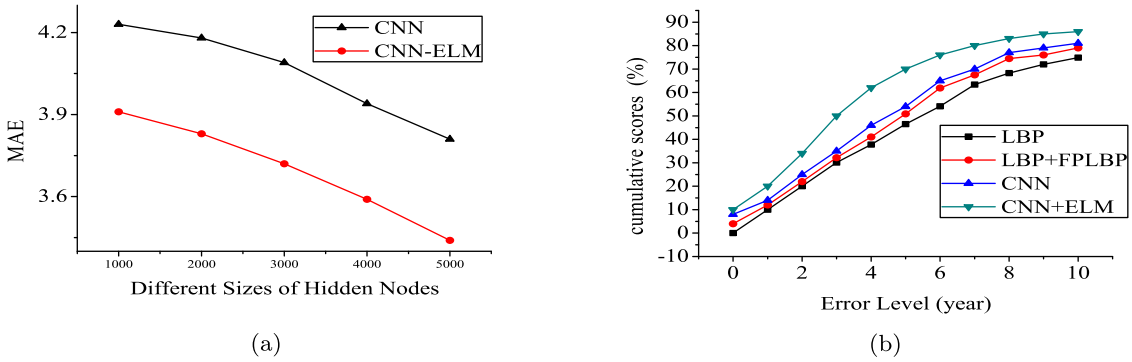


Fig. 7. (a) MAEs of CNN–ELM on MORPH-II under different conditions; (b) the cumulative scores (CS) of age estimation using CNN–ELM.

accuracy of age and gender classification depends on two aspects: feature extractor and classifier. CNN has shown good extraction ability in our hybrid structure. What is more, compared with a plain CNN, due to the same convolutional layers, the feature extractor plays the same role during the experiment, which means the extractor of ELM improves the accuracy of age classification. The phenomenon analysis above has proved that our hybrid structure can make full use of the advantages of CNN and ELM and gain a better performance.

We present the classification accuracy under different hidden layer nodes of ELM classifier in Fig. 10(c). It can be seen from figure below that the classification accuracy ascends fast at the beginning while when the hidden layer nodes are between 2500 and 4000, the system reaches the highest accuracy. Then its accuracy drops sightly when the number of hidden nodes exceed 4000. In general, the accuracy grows with the increasing of hidden layer nodes and then gets the highest accuracy while if the hidden nodes exceed 4000, the error rate rises sightly because of heavy commutation costs, complex computations, and so on.

7.3.3. Age misclassification

From the experimental results, we find that the case of misclassification has happened in our proposed algorithm and we show parts of misclassification results in Fig. 6. The subjects in the top row express that the older are mistakenly classified as the younger while the subjects in the bottom row denote that the younger are mistakenly classified as the older. Unconstrain face images used for our experiments is the main reason of misclassification and we can learn from Fig. 6 that most notable mistakes are caused by blur or low resolution and heavy makeup.

7.4. Age estimation with MORPH-II

In this section, we use the MORPH-II to verify the performance of our CNN–ELM and compare our structure with plain CNN, which includes identical convolutional layers. Fig. 7 presents age estimation and cumulative scores.

Table 4
MAEs of different age estimation algorithms for the MORPH-II database.

Method	MAE
LBP [51]	7.05
BIF [53]	5.09
OHRank [54]	6.07
KPLS [55]	4.04
KCCA [56]	3.98
RED–SVM [57]	6.49
Rank-FFS [58]	4.42
CSOHR [20]	3.82
Plain CNN [2]	3.81
CNN+ELM [ours]	3.44

As seen from Fig. 7(a) above, we can observe that the MAEs decrease as hidden nodes increase and that the performance of proposed structures are better than that of CNN. In the same hidden nodes, our CNN-ELM has the better performance and the MAE of age estimation for our CNN-ELM is 3.44. Fig. 7(b) presents cumulative score (CS) of age estimation. It is apparent that our proposed CNN–ELM outperform other state-of-the-art algorithms by a significant margin.

As seen from Table 4, the listed age estimation approaches acquired different MAEs, while our proposed CNN–ELM achieved the best results. We use the ELM model as the classification model, due to its efficient and fast classification ability, our CNN–ELM obtains lower MAEs than compared algorithms.

7.5. Gender classification using adience benchmark

7.5.1. Error rate under different conditions

In this time, we will show the error rate of gender classification under different iterations in Fig. 8(a). We compare our algorithm with CNN owning the same convolutional layers. The learning rate is the same as applied in age classification. From Fig. 8(a), we can find that with the increase of iterations, the error rate of the two algorithms changes from high to low and our hybrid

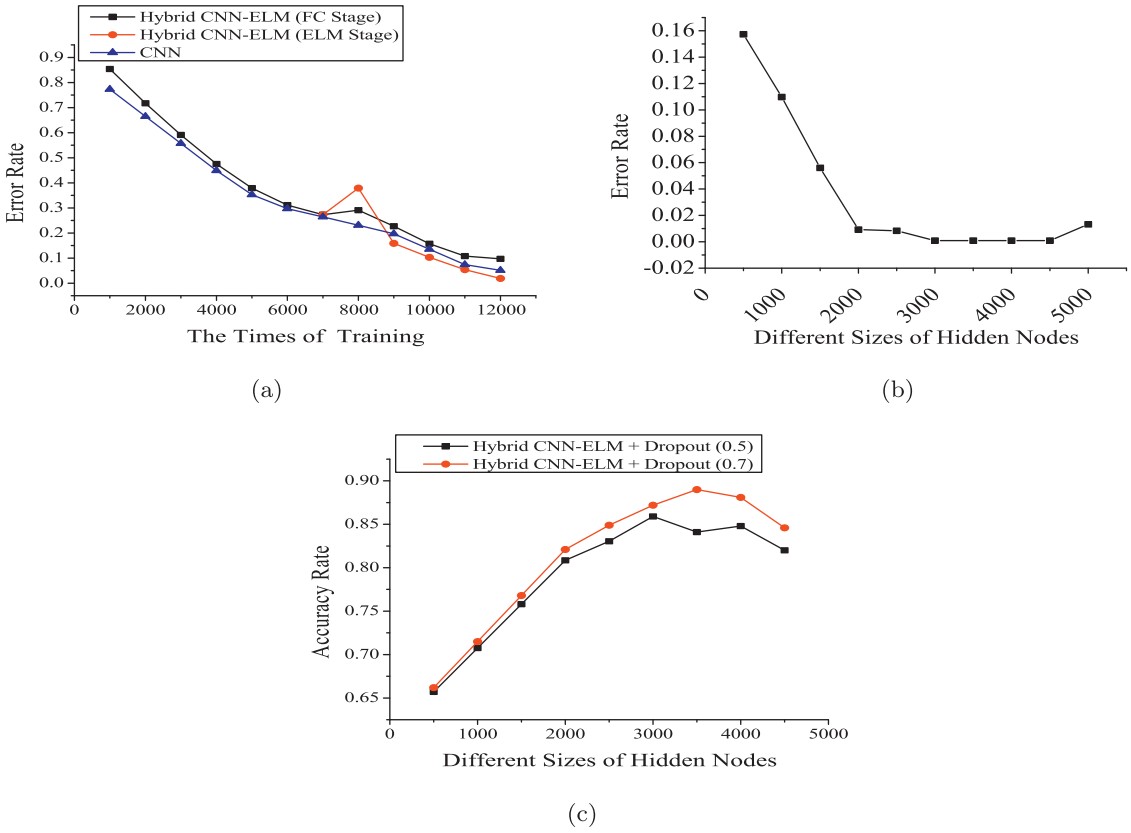


Fig. 8. Experimental results of gender classification on Adience Benchmark. (a) Error rate under different training times; (b) error rate under different sizes of hidden nodes; (c) error rate under different hidden layer nodes.

Table 5
The accuracy of gender classification.

Method	Accuracy
LBP [51]	75.3% ± 0.9%
FPLBP [52]	75.5% ± 0.8%
LBP + FPLBP + Dropout 0.5 [1]	77.8% ± 1.3%
Best from [2]	86.8% ± 1.4%
Proposed CNN–ELM + Dropout 0.5	87.3% ± 1.0%
Proposed CNN–ELM + Dropout 0.7	88.2% ± 1.7%

CNN–ELM during the FC stage gets a higher error rate than a plain CNN while a lower error rate is obtained during the ELM stage. Meanwhile, during the beginning of ELM stage, the error rate has a fluctuate due to the adjustment and adaptation process of hybrid system and our ELM classifier learns faster and gets a lower error rate compared with a plain CNN finally.

Training error rates under different sizes of hidden nodes are presented in Fig. 8(b) and the number of iterations is 12K. With the increase of hidden layer nodes, the error rate declines faster compared with the training process of age classification because gender classification is just a binary task. After the hidden nodes increase to 2000, the error rate nearly tends to 0 and this process lasts until the nodes are 4500. Finally, the error rate begins to fluctuate due to the heavy overfitting. We will set the hidden node as 3500 in the next experiments.

7.5.2. Accuracy of gender classification

Now, we present the accuracy of gender classification with different dropouts compared with algorithms mentioned in age classification experiments in Table 5 and our accuracy includes mean accuracy ± standard deviation. Because gender classification is a binary classification task, the whole process includes training and testing is faster compared with age classification. At the same time, the accuracy of gender classification is higher than that of age classification under the same listed algorithms. More importantly, our proposed algorithm gets the highest accuracy compared with other algorithms. There is no doubt that dropout structures reduce the overfitting and improve the accuracy of proposed system [2].

We show the accuracy of our hybrid system under different hidden layer nodes in Fig. 8(c). The changeable trend of the accuracy is the same as that presented in Fig. 10(a) while the average accuracy of our gender classification is higher than that of age classification because gender estimation is just a binary classification task.

7.5.3. Gender misclassification

We will present some examples of gender misclassification in Fig. 9 and it is noticeable to find that misclassifications frequently happen when the estimation subject is a baby or blur or heavy makeup image. In that case, the male (female) subjects mistakenly classified as a female (male).

8. Gender classification with MORPH-II

In this section, our CNN-ELM is used to classify the gender of human facial images for MOORPH-II. Fig. 10(a) shows the accuracy of gender classification under different hidden nodes. It is can be seen that with the increase of hidden nodes, the classification accuracy increases and our CNN-ELM obtains a better results than plain CNN because of ELM' efficient classification ability. Fig. 10(b) shows ROC curves, which demonstrates the contribution of various components of our CNN–ELM.

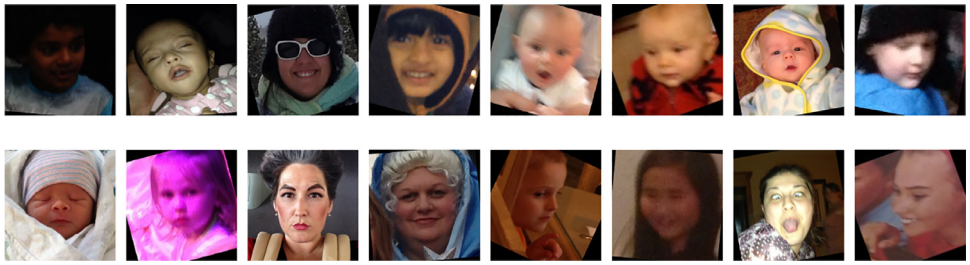


Fig. 9. Gender misclassification.

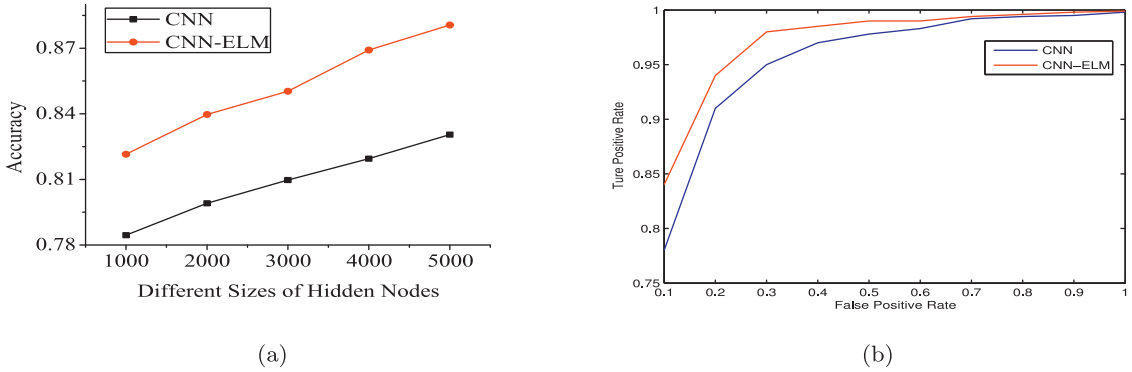


Fig. 10. (a) Accuracy rate under different hidden nodes; (b) ROC curves for gender estimation results.

9. Conclusion

Automatically classifying the unconstrained age and gender tasks is a challenging research topic while few researchers have paid attention on this issue. CNN has shown a perfect feature extraction ability while ELM has been proved to be a powerful classifier. In order to make full use of the advantages of this two structures, we propose a hybrid CNN–ELM structure to process the human facial image tasks. Firstly, we present the hybrid structure in detail including design of parameters and layers, analysis of the hybrid architecture, and the derivation of back-propagation in this system during the iterations. Then we adopt several measures to lower the risk of overfitting, for instance, ELM without tuning the weights and biases owns the ability to overcome overfitting while obtaining a stochastic crop 227×227 pixels from the input images which contain 256×256 pixels also limits the risk of overfitting. Meanwhile, different dropout measures are adopted to do the same works. Finally, we use enough experiments to test the performance of our hybrid CNN–ELM using Adience Benchmark and MORPH-II. Experimental results show that our hybrid algorithm not only accelerates the whole training process, but also improves the accuracy of classification.

Acknowledgments

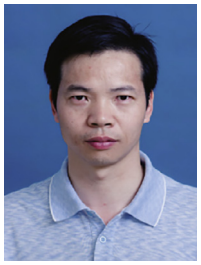
This work was supported in part by the Key Program of National Natural Science Foundation of China under Grant 61432005, in part by the National Outstanding Youth Science Program of National Natural Science Foundation of China under Grant 61625202, in part by the International (Regional) Cooperation and Exchange Program of National Natural Science Foundation of China under Grant 61661146006, in part by the National Natural Science Foundation of China under Grant 61370095 and Grant 61472124, and in part by the International Science & Technology Cooperation Program of China under Grant 2015DFA11240 and Grant 2015AA015303.

References

[1] E. Eidinger, R. Enbar, T. Hassner, Age and gender estimation of unfiltered faces, IEEE Trans. Inf. Forensics Secur. 9 (12) (2014) 2170–2179.
[2] G. Levi, T. Hassncer, Age and gender classification using convolutional neural networks, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 34–42.
[3] Y.H. Kwon, N. da Vitoria Lobo, Age classification from facial images, in: Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. CVPR'94, 1994, pp. 762–767, doi:10.1109/CVPR.1994.323894.
[4] T.R. Kalansuriya, A.T. Dharmaratne, Facial image classification based on age and gender, in: Proceedings of the 2013 International Conference on Advances in ICT for Emerging Regions (ICTer), 2013, pp. 44–50.
[5] D. Svozil, V. Kvasnicka, J. Pospichal, Introduction to multi-layer feed-forward neural networks, Chemometr. Intell. Laborat. Syst. 39 (1) (1997) 43–62.
[6] X.X. Niu, C.Y. Suen, A novel hybrid CNN–SVM classifier for recognizing handwritten digits, Pattern Recogn. 45 (4) (2012) 1318–1325.
[7] S.B. Kim, K.S. Han, H.C. Rim, S.H. Myaeng, Some effective techniques for naive Bayes text classification, IEEE Trans. Knowl. Data Eng. 18 (11) (2006) 1457–1466.
[8] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[9] F.S. Khan, J. van de Weijer, R.M. Anwer, M. Felsberg, C. Gatta, Semantic pyramids for gender and action recognition, IEEE Trans. Image Process. 23 (8) (2014) 3633–3645, doi:10.1109/TIP.2014.2331759.
[10] H. Guang-Bin, C. Lei, S. Chee-Kheong, Universal approximation using incremental constructive feedforward networks with random hidden nodes., IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[11] S. Lawrence, C.L. Giles, A.C. Tsoi, A.D. Back, Face recognition: a convolutional neural-network approach, IEEE Trans. Neural Netw. 8 (1) (1997) 98–113, doi:10.1109/72.554195.
[12] F. Liu, G. Lin, C. Shen, {CRF} learning with {CNN} features for image segmentation, Pattern Recogn. 48 (10) (2015) 2983–2992. http://dx.doi.org/10.1016/j.patcog.2015.04.019. Discriminative Feature Learning from Big Data for Visual Recognition

[13] G.S. Xie, X.Y. Zhang, S. Yan, C.L. Liu, Hybrid CNN and dictionary-based models for scene recognition and domain adaptation, IEEE Trans. Circ. Syst. Video Technol. PP (99) (2015) 1, doi:10.1109/TCSVT.2015.2511543.
[14] J. Tang, C. Deng, G.B. Huang, B. Zhao, Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine, IEEE Trans. Geosci. Remote Sens. 53 (3) (2015) 1174–1185, doi:10.1109/TGRS.2014.2335751.
[15] S. Fu, H. He, Z.G. Hou, Learning race from face: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (12) (2014) 2483–2509, doi:10.1109/TPAMI.2014.2321570.
[16] Y. Fu, G. Guo, T.S. Huang, Age synthesis and estimation via faces: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 1955–1976, doi:10.1109/TPAMI.2010.36.
[17] N. Ramanathan, R. Chellappa, Modeling age progression in young faces, in: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, 2006, pp. 387–394, doi:10.1109/CVPR.2006.187.
[18] X. Geng, Z.H. Zhou, K. Smith-Miles, Automatic age estimation based on facial aging patterns, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) (2007) 2234–2240, doi:10.1109/TPAMI.2007.70733.
[19] G. Guo, Y. Fu, C.R. Dyer, T.S. Huang, Image-based human age estimation by manifold learning and locally adjusted robust regression, IEEE Trans. Image Process. 17 (7) (2008) 1178–1188, doi:10.1109/TIP.2008.924280.
[20] K.Y. Chang, C.S. Chen, A learning framework for age rank estimation based on face images with scattering transform, IEEE Trans. Image Process. 24 (3) (2015) 785–798, doi:10.1109/TIP.2014.2387379.
[21] C. Li, Q. Liu, W. Dong, X. Zhu, J. Liu, H. Lu, Human age estimation based on locality and ordinal information, IEEE Trans. Cybern. 45 (11) (2015) 2522–2534, doi:10.1109/TCYB.2014.2376517.
[22] A. Lanitis, The FG-net aging database, 2002 http://www-prima.inrialpes.fr/FGnet/html/benchmarks.html.
[23] K. Ricanek, T. Tesafaye, Morph: a longitudinal image database of normal adult age-progression, in: Proceedings of the Seventh International Conference on Automatic Face and Gesture Recognition, 2006. FGR 2006, 2006, pp. 341–345, doi:10.1109/FGR.2006.78.
[24] G. Guo, X. Wang, A study on human age estimation under facial expression changes, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2547–2553, doi:10.1109/CVPR.2012.6247972.
[25] E. Makinen, R. Raisamo, Evaluation of gender classification methods with automatically detected and aligned faces, IEEE Trans. Pattern Anal. Mach. Intell. 30 (3) (2008) 541–547, doi:10.1109/TPAMI.2007.70800.
[26] B.A. Golomb, D.T. Lawrence, T.J. Sejnowski, Sexnet: a neural network identifies sex from human faces, in: Proceedings of the 1990 Conference on Advances in neural information processing systems, 3, 1990, pp. 572–577.
[27] B. Moghaddam, M.-H. Yang, Learning gender with support faces, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 707–711, doi:10.1109/34.1000244.
[28] S. Baluja, H.A. Rowley, Boosting sex identification performance, Int. J. Comput. Vis. 71 (1) (2006) 111–119, doi:10.1007/s11263-006-8910-9.
[29] M. Toews, T. Arbel, Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion, IEEE Trans. Pattern Anal. Mach. Intell. 31 (9) (2009) 1567–1581, doi:10.1109/TPAMI.2008.233.
[30] S. Yu, T. Tan, K. Huang, K. Jia, X. Wu, A study on gait-based gender classification, IEEE Trans..Image Process. 18 (8) (2009) 1905–1910, doi:10.1109/TIP.2009.2020535.
[31] J. Bekios-Calfa, J.M. Buenaposada, L. Baumela, Revisiting linear discriminant techniques in gender recognition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (4) (2011) 858–864, doi:10.1109/TPAMI.2010.208.
[32] H. Chen, A. Gallagher, B. Girod, Face modeling with first name attributes, IEEE Trans. Pattern Anal. Mach. Intell. 36 (9) (2014) 1860–1873, doi:10.1109/TPAMI.2014.2302443.
[33] H. Han, C. Otto, X. Liu, A.K. Jain, Demographic estimation from face images: human vs. machine performance, IEEE Trans. Pattern Anal. Mach. Intell. 37 (6) (2015) 1148–1161, doi:10.1109/TPAMI.2014.2362759.
[34] Y. Lcun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[35] F. Jialue, X. Wei, W. Ying, G. Yihong, Human tracking using convolutional neural networks, IEEE Trans. Neural Netw. 21 (10) (2010) 1610–1623.
[36] Y. Cao, Y. Chen, D. Khosla, Spiking deep convolutional neural networks for energy-efficient object recognition, Int. J. Comput. Vis. 113 (1) (2015) 54–66.
[37] M. Duan, K. Li, X. Liao, K. Li, A parallel multiclassification algorithm for big data using an extreme learning machine, IEEE Trans. Neural Netw. Learn. Syst. PP (99) (2017) 1–15, doi:10.1109/TNNLS.2017.2654357.
[38] B. Zuo, G.B. Huang, D. Wang, W. Han, M.B. Westover, Sparse extreme learning machine for classification, IEEE Trans. Cybern. 44 (10) (2014) 1858–1870.
[39] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (2) (2012) 513–529, doi:10.1109/TSMCB.2011.2168604.
[40] Y. Yang, Q.M. Wu, Y. Wang, K.M. Zeeshan, X. Lin, X. Yuan, Data partition learning with multiple extreme learning machines, IEEE Trans. Cybern. 45 (6) (2014) 1463–1475.
[41] J. Luo, C.M. Vong, P.K. Wong, Sparse Bayesian extreme learning machine for multi-classification., IEEE Trans. Neural Netw. Learn. Syst. 25 (4) (2014) 836–843.
[42] J. Shuiwang, Y. Ming, Y. Kai, 3d convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231.
[43] Z. Dong, Y. Wu, M. Pei, Y. Jia, Vehicle type classification using a semisupervised convolutional neural network, IEEE Trans. Intell. Transp. Syst. 16 (4) (2015) 1–10.
[44] Y. Yang, Y. Wang, X. Yuan, Bidirectional extreme learning machine for regression problem and its learning effectiveness, IEEE Trans. Neural Netw. Learn. Syst. 23 (9) (2012) 1498–1505, doi:10.1109/TNNLS.2012.2202289.
[45] P. Sermanet, Y. Lecun, Traffic sign recognition with multi-scale convolutional networks, in: Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN)„ 2011, pp. 2809–2813.
[46] N. Pinto, D.D. Cox, J.J. Dicarlo, Why is real-world visual object recognition hard? PLoS Comput. Biol. 4 (1) (2008) 86–89.
[47] K. He, J. Sun, Convolutional neural networks at constrained time cost, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5353–5360, doi:10.1109/CVPR.2015.7299173.
[48] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2) (2012) 2012.
[49] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675–678.
[50] Open university of israel, http://www.openu.ac.il/home/hassner/Adience.
[51] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041, doi:10.1109/TPAMI.2006.244.
[52] L. Wolf, T. Hassner, Y. Taigman, Descriptor based methods in the wild, Proceedings of the post-ECCV Faces in Real-Life Images Workshop, 2008.
[53] G. Guo, G. Mu, Y. Fu, T.S. Huang, Human age estimation using bio-inspired features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, 2009, pp. 112–119.
[54] K.Y. Chang, C.S. Chen, Y.P. Hung, Ordinal hyperplanes ranker with cost sensitivities for age estimation, in: Proceedings of the CVPR 2011, 2011, pp. 585–592, doi:10.1109/CVPR.2011.5995437.
[55] G. Guo, G. Mu, Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression, in: Proceedings of the CVPR 2011, 2011, pp. 657–664, doi:10.1109/CVPR.2011.5995404.
[56] G. Guo, G. Mu, Joint estimation of age, gender and ethnicity: CCA vs. PLS, in: Proceedings of the 2013 Tenth IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2013, pp. 1–6, doi:10.1109/FG.2013.6553737.
[57] K.Y. Chang, C.S. Chen, Y.P. Hung, A ranking approach for human ages estimation based on face images, in: Proceedings of the 2010 twentieth International Conference on Pattern Recognition, 2010, pp. 3396–3399, doi:10.1109/ICPR.2010.829.
[58] Y.L. Chen, C.T. Hsu, Subspace learning for facial age estimation via pairwise age ranking, IEEE Trans. Inf. Forensi. Secur. 8 (12) (2013) 2164–2176, doi:10.1109/TIFS.2013.2286265.

Mingxing Duan is working toward the Ph.D. degree in the School of Computer Science at National University of Defense Technology, China. His research interest includes Big Data, Machine Learning.



Kenli Li received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2003. He was a Visiting Scholar with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, from 2004 to 2005. He is currently a Full Professor of Computer Science and Technology with Hunan University, Changsha, China, and also the Deputy Director of the National Supercomputing Center, Changsha. He has authored over 150 papers in international conferences and journals, such as the IEEE-TC, the IEEE-TPDS, and the IEEE-TSP. His current research interests include parallel computing, cloud computing, and big data computing. He is an outstanding member of CCF. He is currently serves on the editorial boards of the IEEE TRANSACTIONS ON COMPUTERS and the International Journal of Pattern Recognition and Artificial Intelligence.

Canqun Yang received the M.S. and Ph.D. degrees in Computer Science from the National University of Defense Technology, China, in 1995 and 2008, respectively. Currently he is a Professor at the National University of Defense Technology. His research interests include programming languages and compiler implementation. He is the major designer dealing with the compiler system of the Tianhe Supercomputer.

Keqin Li is a SUNY Distinguished Professor of computer science. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multi-core computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber physical systems. He has published over 440 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, Journal of Parallel and Distributed Computing. He is an IEEE Fellow.