Received May 18, 2018, accepted June 15, 2018, date of publication June 19, 2018, date of current version July 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2848966

Multiple Features With Extreme Learning Machines For Clothing Image Recognition

RUIFAN LI[1,2], (Member, IEEE), WENCONG LU[1], HAOYU LIANG[1], YUZHAO MAO[1], AND XIAOJIE WANG[1,2]

[1]School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Engineering Research Center of Information Networks, Ministry of Education, Beijing 100876, China

Corresponding author: Ruifan Li (rfli@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61273365, Grant 61472046, and Grant 61472048, in part by the National Social Science Fund of China under Grant 2016ZDA055, and in part by the Discipline Building Plan in 111 Base under Grant B08004.

ABSTRACT Clothing image recognition has recently received considerable attention from many communities, such as multimedia information processing and computer vision, due to its commercial and social applications. However, the large variations in clothing images' appearances and styles and their complicated formation conditions make the problem challenging. In addition, a generic treatment with convolutional neural networks (CNNs) cannot provide a satisfactory solution considering the training time and recognition performance. Therefore, how to balance those two factors for clothing image recognition is an interesting problem. Motivated by the fast training and straightforward solutions exhibited by extreme learning machines (ELMs), in this paper, we propose a recognition framework that is based on multiple sources of features and ELM neural networks. In this framework, three types of features are first extracted, including CNN features with pre-trained networks, histograms of oriented gradients and color histograms. Second, those low-level features are concatenated and taken as the inputs to an autoencoder version of the ELM for deep feature-level fusion. Third, we propose an ensemble of adaptive ELMs for decision-level fusion using the previously obtained feature-level fusion representations. Extensive experiments are conducted on an up-to-date large-scale clothing image data set. Those experimental results show that the proposed framework is competitive and efficient.

INDEX TERMS Clothing image recognition, extreme learning machines, feature fusion, autoencoder ELM, ensemble learning.

I. INTRODUCTION

Abundant clothing images are easily available from e-commercial platforms, such as amazon.com in the USA and taobao.com in China. Analyzing clothing images has attracted researchers from multimedia information processing and computer vision for the past several years [1]–[3] due to this analysis' primary importance for commercial and social applications. For example, clothing recognition is beneficial for identifying an individual in a personal photo collection. As another example, he or she could automatically annotate his or her travel photo with recognized clothing types and detailed attributes and later share them with his or her friends. One of those important clothing-related analysis tasks, clothing image recognition has received considertalbe attention.

The clothing image recognition problem can technically be formulated as a classification problem. However, different from the generic task of object classification or recognition, clothing recognition from still images has its own characteristics, which makes this task even more challenging. To be specific, the clothing appearances have large variations in texture, color, and style. Additionally, the clothing is usually not rigid, and it could show different geometric appearances, even for the same clothes. In addition, the formation conditions of clothing images could largely vary, such as from outdoors to professional indoor shootings.

In general, the approaches to recognizing clothing images can be grouped into two categories. Earlier work [1], [4] on clothing recognition has mainly depended on hand-crafted features, such as histograms of oriented gradients (HOG) [5],

scale-invariant feature transforms (SIFT) [6], and color histograms. Although some advances have been achieved by those methods, there is still room for improvement. Recently, since 2012 deep learning especially, convolutional neural networks (CNNs) have been proposed. Those networks have been widely applied in tasks of large-scale image classification and have achieved significant performance [7]. Generic neural networks for clothing image recognition could give a solution with high accuracy. However, the features learned from those neural networks are not easily understandable. Furthermore, the training for such deep neural networks is time-consuming, and the tuning of the network parameters could be highly difficult. Therefore, despite the remarkable progress that has been made in clothing recognition, the training time and recognition performance still have room for improvement.

In this paper, we explore multiple features and propose a framework for clothing image classification that is based on a type of random neural network, extreme learning machines [8] (ELMs, to be introduced in detail in Section III). In this framework, for an image, three types of features are first extracted, which are the CNN features from pre-trained networks, HOG features, and color histograms. Second, those low-level representations are concatenated and are taken as the inputs to an Autoencoder variant of ELM (AE-ELM) to obtain a type of high-level fusion feature. Third, we propose an ensemble strategy with ELMs, known as Ada-ELMs, to classify clothing images with previously obtained high-level image representations through the AE-ELM. To evaluate our framework, we conduct extensive experiments on an up-to-date and publicly available dataset, DeepFashion [2]. The experimental results demonstrate that our proposed framework is competitive and fast in the task of clothing image recognition.

The remainder of this article is organized as follows. We first review the related work in Section II. Next, in Section III we propose our framework for recognizing clothing from still images, including our designed CNN feature extraction, HOG feature extraction, and color histogram extraction. In addition, the general ELM, its variant AE-ELM, and our proposed Ada-ELMs are illustrated in detail. In Section IV, we report our experiments and their results. Lastly, we present the study's conclusions and outline several future research directions in Section V.

II. RELATED WORK

Due to the increasingly large business value of the fashion and shopping industry, automatic clothing image analysis has received considerable attention. One trend is to use attribute learning to give a fine-grained description of a clothing image, which has recently been widely applied in the computer vision community [1], [9]–[16].

However, one of the major challenges that is faced with attribute learning is the lack of well-labeled training data because of the heavy cost of the labor and time. In addition, setting up these attributes usually requires domain-specific knowledge, which can then be used to label the data. To overcome this difficulty, Berg *et al.* [11] proposed to discover the attributes and visual appearance by mining the descriptive text of images from the Internet. For clothing images, Chen *et al.* [1] focus on learning the visual attributes of clothing on the human upper body only. Recently, Shankar *et al.* [13] proposed to discover all of the attributes that are present in an image in a weakly supervised scenario based on deep neural networks. In general, those studies take attribute learning as a separate task. However, for large-scale clothing images, the attributes are highly related to the clothing category, and those categories cannot be ignored while detecting attributes.

Another line of research on analyzing clothing images is based on methods from pose estimation and person detection [17]–[21]. Clothing parsing, a recently proposed task, is to predict a semantic category, such as shirt, skirt, and shoes, for each pixel in an image. The parsing results could then be further used for the clothing recognition. Most notably, Liu *et al.* [17] address the cross-scenario application problem in which a daily human photo is considered to retrieve a clothing shop photo. They alleviate the discrepancy of those two distributions by using a sparsely coded transfer matrix. Kalantidis *et al.* [19] also consider a similar cross-scenario approach, where they start from pose estimation and then utilize clothing parsing. Recently, Yamaguchi *et al.* [20] propose an unconstrained clothing parsing without user-provided tag information for clothing retrieval. The insight obtained from those methods is the use of body pose estimation for clothing parsing. However, the performances of those approaches largely rely on an accurate pose estimation and human parts detection and cannot easily extend to the large-scale clothing parsing and recognition problem.

Over past several years, deep learning, which is motivated by the biological deep and distributed structure of the human brain, has been proposed to learn hierarchical and effective representations to facilitate various tasks in computer vision, from the year 2006 [22], [23]. The basic idea of deep learning methods is that they use some simple non-linear neural neurons to compositionally build a very complex fitting function. To name a few such methods, deep learning methods, especially supervised convolutional neural networks (CNNs) [24], [25], unsupervised autoencoders, restricted Boltzmann machines, and generative adversarial networks [26] have been successfully applied due to the availability of computational power and the volume of data in large-scale image classification [7], [27], [28], multi-modal information processing [29]–[32], and object detection [33].

Deep learning also has an advantage for multi-task learning, which aims to achieve better performance by simultaneously exploring multiple closely related tasks. Deep learning methods can learn deep representations that capture those underlying factors. Because of this natural connection, multi-task learning could then be a possible means for large-scale clothing image analysis. Very recently, several methods

based on deep learning for multi-task learning have been proposed [34]–[38]. Notably, Zhang *et al.* [34] proposed to combine parts-based models and CNNs for deep feature representation, to obtain an attribute description for people under the framework of a multi-task problem. However, this PANDA framework is specifically designed for relatively small-scale datasets and cannot easily be extended to large-scale problems in the real world. Bai *et al.* [35] proposed multi-task deep neural networks for text-based image retrieval. In the proposed framework, query-sharing layers for image representation and query-specific layers for relevance estimation are learned jointly. In general, the representation power of CNNs compared with the shallow hand-crafted visual features, such as HoG and SIFT, provides insights into how to make learning multiple tasks possible. However, the performance of those partly or fully connected neural networks heavily relies on the quality of the data labels. In addition, those CNN-based methods do not consider the correlation between attributes or the cross category of visual attributes, especially for large-scale clothing datasets that require high-speed computational power.

III. PROPOSED FRAMEWORK

We describe our proposed framework for clothing image recognition in this section. First, the key components, ELMs and AE-ELMs, including their concepts and basic ideas and learning algorithms, are reviewed in sections III-A and III-B. Second, three adopted features are introduced. Third, our framework, which uses those three features and ELMs, is described.

A. EXTREME LEARNING MACHINES

An Extreme Learning Machine (ELM) [8], [39]–[44] is originally proposed for learning single hidden layer feedforward networks (SLFNs). Specifically, the output of an SLFN with $\tilde{L}$ hidden units can be represented as

$$f_{\tilde{L}}(\mathbf{x}) = \sum_{l=1}^{\tilde{L}} \boldsymbol{\beta}_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{1}$$

in which the input vector $\mathbf{x}$ resides in a D dimensional space, i.e., $\mathbf{x} \in \mathbb{R}^D$, and $h_i(\mathbf{x})$ is the output of the i-th hidden unit. Compactly, the network outputs $\mathbf{h}(\mathbf{x})$ are vectorized as $\left[h_1(\mathbf{x}), \cdots, h_{\tilde{L}}(\mathbf{x})\right]$, and the weights $\boldsymbol{\beta}$ between the hidden units and the output units are represented as a matrix $\left[\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_{\tilde{L}}\right]^T$. Note that the superscript T stands for the operation of matrix transpose. Significantly, an ELM differs from the traditional multi-layer perceptron (MLP) as follows. Each hidden unit has the activation function $h(\boldsymbol{x}) = h(\mathbf{x}; \boldsymbol{a}, b)$. With regard to the additive units, the activation function is defined as $h(\boldsymbol{a}\cdot\boldsymbol{x}+b)$, in which the parameters $\boldsymbol{a}$ and b are kept fixed once they are randomly initialized. Other functions, such as the radial basis function $h(b||\boldsymbol{x} - \boldsymbol{a}||)$, can also be used for constructing the neural hidden units.

Moreover, the learning for the weights $\boldsymbol{\beta}$ usually does not adopt an iterative process but instead has a direct single-step solution. To be specific, when given a training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, in which the data vector $\boldsymbol{x}_i \in \mathbb{R}^D$ and its target vector $\boldsymbol{y}_i \in \mathbb{R}^d$, an ordinary ELM attempts to solve the following optimization objective,

$$\arg\min_{\boldsymbol{\beta}} ||\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{H}\boldsymbol{\beta} - \boldsymbol{Y}||_2^2 \tag{2}$$

where $||\cdot||_2$ stands for ℓ_2-norm, and the λ is a positive constant used for achieving better stability and generalization. Moreover, the matrix $\boldsymbol{H}$ of its hidden layer outputs is

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\boldsymbol{x}_1) & \cdots & h_{\tilde{L}}(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\boldsymbol{x}_N) & \cdots & h_{\tilde{L}}(\boldsymbol{x}_N) \end{bmatrix}. \tag{3}$$

And the target matrix $\boldsymbol{Y}$ is

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1^T \\ \vdots \\ \boldsymbol{y}_N^T \end{bmatrix} = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \ddots & \vdots \\ y_{N1} & \cdots & y_{Nm} \end{bmatrix}. \tag{4}$$

According to the theory of ELM [40], when the parameter λ is infinitely large, the above objective function (2) can be solved as using the minimal norm least square method. Thus, the output weight matrix

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{Y} \tag{5}$$

in which $\boldsymbol{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\boldsymbol{H}$. Furthermore, through the orthogonal projection method, the Moore-Penrose generalized inverse is calculated as $\boldsymbol{H}^{\dagger} = \boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{H}^T\right)^{-1}$, if the matrix $\boldsymbol{H}\boldsymbol{H}^T$ is nonsingular. Otherwise, the inverse matrix $\boldsymbol{H}^{\dagger} = \left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T$, if the matrix $\boldsymbol{H}^T\boldsymbol{H}$ is nonsingular.

With the minimum norm of its output weights $\boldsymbol{\beta}$, this ELM achieves better generalization performance. The weights from the hidden layer to the output layer can then be given as

$$\boldsymbol{\beta} = \boldsymbol{H}^T\left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H}\boldsymbol{H}^T\right)^{-1}\boldsymbol{Y} \tag{6}$$

or

$$\boldsymbol{\beta} = \left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Y}. \tag{7}$$

To summarize, the algorithm on ELM learning for a general SLFN is given in Algorithm 1. Huang et. al. [39] show that for an ELM, the solutions given by equation (6) and equation (7) provide a unified learning for regression and classification problems. Additionally, the ELM gains a higher generalization performance and faster learning speed.

B. AUTOENCODER EXTREME LEARNING MACHINES (AE-ELMs)

The autoencoder is a special version of multi-layer perceptron [45]. The aim of the autoencoder network is to learn a compact or a sparse representation of a set of data in an unsupervised fashion. In this setting, the output and input layers

Algorithm 1 The ELM Learning Algorithm

1: Generate the weights and biases between the input layer and hidden layer with some continuous distribution, e.g., the standard Gaussian distribution $\mathcal{N}(0, 1)$;
2: Calculate the output matrix $\boldsymbol{H}$ of the hidden layer using equation (3);
3: Estimate the weights $\boldsymbol{\beta}$ of the output layer using equation (6) or equation (7).

are set to the same values. Next, a general back-propagation algorithm is applied for training this network.

To extend the representation capacity of the ELM, Kasun *et al.* [46] proposes a simple but effective method for obtaining the autoencoder version of the unsupervised ELM, known as AE-ELMs. Specifically, the general ELM is modified as follows. 1) The output neurons are set to the same values as the input neurons. 2) The random weights and biases of the hidden neurons are intentionally chosen to be orthogonal. Next, we formalize the AE-ELMs.

In an AE-ELM, the orthogonal weights and biases project the input data into a representation space with dimensionality L and are computed as follows

$$\boldsymbol{h}(\boldsymbol{x}) = g(\boldsymbol{a}^T\boldsymbol{x} + \boldsymbol{b}). \tag{8}$$

Note that the weights and biases are subjected to $\boldsymbol{a}^T\boldsymbol{a} = \boldsymbol{I}$ and $\boldsymbol{b}^T\boldsymbol{b} = 1$. Here, $\boldsymbol{I}$ is the identity matrix. The weight matrix $\boldsymbol{a} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_L]$ is composed of L orthogonal random vectors. The bias vector $\boldsymbol{b} = [b_1, \cdots, b_L]$ is an orthogonal random vector. Note that the function $g(\cdot)$ operates on each element of a vector. According to the Johnson-Lindenstrauss lemma, the characteristic of being universal approximators are guaranteed and proven for AE-ELMs. Then, the output weights $\boldsymbol{\beta}$ of the AE-ELMs form the mapping from the representation space to the input space. For a compact and sparse representation space, the weights $\boldsymbol{\beta}$ of AE-ELM can be calculated as in equation (6) and (7) but with the target matrix $\boldsymbol{Y}$ is replaced with the input matrix $\boldsymbol{X}$. In other words, the input matrix $\boldsymbol{X}$ equals $[\mathbf{x}_1, \cdots, \mathbf{x}_N]$, which is also the output of those data as a general autoencoder.

Furthermore, through singular value decomposition, i.e., SVD we have the following equation,

$$\boldsymbol{H}\boldsymbol{\beta} = \sum_{i=1}^{N} \boldsymbol{u}_i \frac{d_i^2}{d_i^2 + \lambda} \boldsymbol{u}_i^T \boldsymbol{X} \tag{9}$$

in which, the vector $\boldsymbol{u}_i$ is the ith eigenvector of matrix $\boldsymbol{H}\boldsymbol{H}^T$ with the corresponding eigenvalue d_i. Evidently, the projection from the input space $\boldsymbol{X}$ to the representation space $\boldsymbol{H}$ is performed through a sigmoid activation function. Therefore, the output weight vector $\boldsymbol{\beta}$ represents the features that are hidden in the data by singular values. To summarize, we note that the output weights in an AE-ELM can be determined analytically, unlike traditional autoencoders, which require some iterative algorithms. In addition, an AE-ELM learns to represent features via singular values, unlike traditional autoencoders, where the actual representation of the data is obtained.

C. FEATURE EXTRACTION

1) CNN FEATURE EXTRACTION

We use the following CNN network for feature extraction. The features extracted from this network are used as our first type of feature. This network has ten layers. An image is taken as input to two consecutive convolution operations with a convolutional kernel size of 3×3 and 64 kernels. Next, a max-pooling operation with the pooling size of 2×2 is applied. Subsequently, the same architecture of two consecutive convolution operations and a max-pooling operation is applied. In this instance, the convolutional kernel has size 3×3 with 128 kernels. Subsequently, this network consists of two fully connected layers with 512 units. The dropout operation is applied for those fully connected layers, with a dropout probability of 0.5. The final output takes on the softmax function with output probabilities to which each category belongs. After being trained, the output of the eighth layer being fully connected in this CNN network is chosen as the features of the input image. The extracted features possess the dimensionality of 512. This CNN model is taken as our baseline for experimental evaluations. In addition, based on this CNN model, we will build other models with multiple feature fusion and decision fusion.

2) HOG FEATURE EXTRACTION

The second feature type adopted in our framework is the HOG features. Historically, the HOG descriptor was originally proposed for the task of human detection [5]. The HOG descriptor has been extensively applied in computer vision and image processing. In this section, we merely review the basic idea and illustrate its motivation for our task of clothing image recognition.

The basic idea of the HOG descriptor is that the appearance and shape of local objects within an image can effectively be captured by the distribution of intensity gradients. The procedure for extracting HOG descriptors usually contains the following steps. The image is first divided into cells with small connected regions. Next, a histogram of gradient directions is collected for all of the pixels of the small cell. Lastly, those histograms are concatenated to obtain the HOG descriptor. The HOG descriptor has certain characteristics, that are beneficial to clothing image recognition. The HOG extraction is performed on local cells, which makes it invariant to photometric and geometric transformations. Furthermore, the three improvements, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization, enhance the robustness of the recognition.

In our experiments, after having been preprocessed to be 50×50, each of the images undergoes the procedure of HOG extraction. During this process, the three parameters are set as follows. The size of a cell is set to 6×6. The cell size of the block is set to 3×3, and the number of orientation classes

is fixed to eight. Subsequently, we obtain the HOG descriptor with the dimensionality of 628.

3) COLOR HISTOGRAM EXTRACTION

The third feature type adopted in our framework is the color histogram. The color histogram represents the distribution of colors in an image. It represents not only the object color and illumination but also relates to the surface roughness and image geometry. Thus, the histogram provides an improved estimate of the illumination and object color. Practically, the histogram is merely obtained by counting pixels that have each possible color. That color information is faster to compute and be applied in a real-time system. In our experiments, we use the histogram of the grayscale image conversion, $v = 0.3r + 0.59g + 0.11b$. This mapping from the RGB space obtains a histogram of dimensionality 256 within the range of [0, 255].

D. ADA-ELMs

The proposed framework is shown in Figure 1. The input image is processed in the following steps. We cropped out the clothing part of each image using the annotation information in the dataset. Then, three types of features, CNN, HOG and Color histogram, are extracted in parallel. For the CNN feature, we extract them from the pre-trained networks. Next, a naive feature-level fusion by concatenation is applied. To acquire the deep fusion, we adopt the AE-ELM with unsupervised learning. Then, the fusion representation of those three features is input to an ensemble classifier, Ada-ELMs. Lastly, the category of the considered image is given by this system.

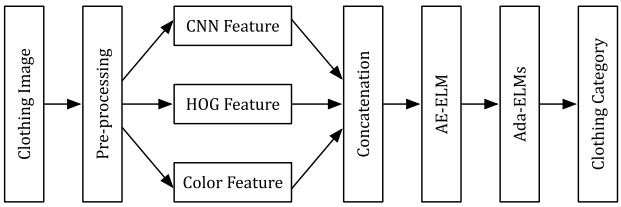


FIGURE 1. The proposed Framework for Clothing Image Classification.

In the previous sections, we have illustrated those three feature extraction methods and the fusion model, AE-ELM. Next, we will describe our ensemble model, Ada-ELMs. Specifically, we fuse the decision results of multiple ELM classifiers. An adaptive fusion algorithm, called Ada-ELMs, is proposed. The idea is to train simultaneously multiple ELM classifiers and assign adaptively the decision weights of those trained ELM classifiers by taking advantage of ELM high-speed learning. The fusion decision based on multiple classifiers can increase the accuracy of the final decision. Specifically, the Ada-ELMs is composed of K independent ELMs with the same number of hidden neurons and with the same activation function. Therefore, the hyper-plane for the training set is optimal. The biases and weights are randomly initialized at the input layer, and the weights at the output layer are calculated directly from a matrix computation. Therefore, for the same training set, the obtained weights differ from one another. It should be noted that the solution of an ELM is globally optimal. Different training sets will result in different solutions. In other words, for an ELM that uses N samples for training, we obtain the first trained ELM. We then train another ELM from those misclassified samples, and the second ELM is optimal for those misclassified samples. If we assign weights for those two ELMs when making a decision, better results could be obtained by correcting some errors that occurs with the first trained ELM classifier. To summarize, the learning algorithm of Ada-ELMs is given in Algorithm 2.

Algorithm 2 Our Ada-ELMs Learning Algorithm

Require: Total samples, N; Total categories, C; Total ELMs, K; Number of hidden neurons of ELMs, L; Activation function of ELMs, $g(\boldsymbol{x}; \boldsymbol{a}, b)$; Weight vector F_w for ELMs with the dimensionality of K; Weight vector S_w for samples with the dimensionality of N, initialized with $\mathbf{1}$ vector.

Ensure: Multiple ELM classifiers with assigned decision weights.

1: Assign initial value for the counter $k = 1$.
2: **repeat**
3: Initialize the kth ELM.
4: Collect the training samples for the kth ELM as follows. IF k equals 1, all samples in the training set are used, ELSE obtain the misclassified $E^{(k-1)}$ samples using the $(k-1)$th ELM, and then obtain the training set $T^{(k)}$ by mixing $E^{(k-1)}$ samples with randomly chosen $\rho N - E^{(k-1)}$ samples from the training set. The coefficient ρ is set to 0.8 in our experiments.
5: Train the kth ELM.
6: Update the sample weights as follows. For every sample in the training set, IF the error occurs with nth sample using the kth ELM, THEN its weight is updated with $S_w^{(n)} = S_w^{(n)} + \frac{\eta}{K}$. The coefficient η is set to 5.0 in our experiments.
7: Compute the decision weight $F_w^{(k)}$ for the kth ELM with the ratio, sum weight of correctly classified samples to that of all samples.
8: Update the counter $k = k + 1$.
9: **until** the counter $k > K$.

Next, during the test phase, for an unknown sample the learned ensemble Ada-ELM gives its final decision by choosing the category that has the largest score. The formula for this fusion decision is shown as follows,

$$\arg\max_{c} \sum_{k=1}^{K} F_w^{(k)} O_c^{(k)} \tag{10}$$

in which, $\boldsymbol{O}^{(k)}$ is the output of the kth ELM, the subscript c is for the category, and F_w is the decision weight of the ELMs.

FIGURE 2. Example images from eight categories, "Dress", "Jeans", "Joggers", "Shorts", "Skirts", "Sweaters", "Tank", and "Tee" in the DeepFashion Dataset. Each column is corresponding to each category.

IV. EXPERIMENTS

To show the effectiveness of our proposed framework for clothing image classification, we conducted extensive experiments on an up-to-date clothing image dataset, DeepFashion. In this section, we will first introduce this publicly available DeepFashion dataset. Thereafter, the results of those methods compared in our experiments are given. Finally, the experimental results and their computational time are reported.

A. DATASET

We evaluate our methods on a publicly available dataset. The up-to-date DeepFashion dataset [2] has been recently collected and organized by the Chinese Hong Kong University.[1] DeepFashion provides the largest number of images and their corresponding annotations. Generally, the DeepFashion dataset is composed of four subsets, each of which is specifically collected and cleaned for some tasks related to analyzing clothing images. One of those four subsets, called Category and Attribute Prediction, can be used for category prediction on clothing images. This image subset has fifty categories and 300,000 clothing images. In addition, each image is annotated with the position by four coordinates of its contained clothing. This facilitates our task of clothing classification without an additional algorithm for clothing detection. Additionally, note that the data in all of those fifty categories are extremely imbalanced. Some categories have more than fifty thousand images, and some categories have no more than hundreds or only a few dozens. The class imbalance problem is out of our focus and is not considered in this paper. Therefore, we choose eight categories from the DeepFashion dataset for our experiments. Those eight categories and their respective quantities of samples are listed as follows: "Dress", "Jeans", "Joggers", "Shorts", "Skirts", "Sweaters", "Tank", and "Tee". Four other categories, including "Cape", "Nightdress", "Shirtdress" and "Sundress", have been merged into "Dress" by the DeepFashion creators. Examples from those eight categories in the subset of DeepFashion are shown in Figure 2. In this figure, the annotated coordinates are not shown. Those images have different sizes. In addition, we have resized each image into 50×50 pixels. We then follow the evaluation annotation of this dataset. On average, we have a total of 20,167 images for each category, 16,948 for training and 3,219 for testing. The data split on the training and testing distribution for the DeepFashion Dataset is summarized in Table 1.

[1] http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html

B. METHODS AND SETTINGS

We adopt the extracted features using CNN networks combined with a multi-layer perceptron, i.e., MLP as the baseline. The categorial cross-entropy function is utilized as the loss

TABLE 1. Class labels and train-test distribution of samples for the deepfashion dataset.

#	Class	Training	Testing	Total
1	Dress	52,138	9,968	62,106
2	Jeans	5,187	997	6,184
3	Joggers	3,260	575	3,835
4	Shorts	14,976	2,931	17,907
5	Skirt	10,794	1,933	12,727
6	Sweater	9,936	1,855	11,791
7	Tank	11,883	2,265	14,148
8	Tee	27,411	5,227	32,638

function for training. Experimentally, we set the parameters as follows: the learning rate is 0.01, the momentum 0.95, the decay of the learning rate 0.00018, and the batch size 50. The nesterov is chosen as the momentum method. We record the changing losses and accuracies during the training procedure in the DeepFashion dataset. After being trained with 40 epochs, the loss decreases to 0.644, and the accuracy achieves 76.5%. The training curve of the losses and accuracies are shown in sub-figures 3a and 3b, respectively.

In general, we compare four groups of methods, which are described as follows. The first group of methods compared comprises the original three features independently associated with two neural classifiers, MLP and ELM. Those two classifiers with different hidden neurons are evaluated. The second group of methods compared comprises naive fusion of three features with an MLP classifier. Neural networks with different hidden neurons are evaluated. The third group of methods compared comprises fusions of three features using AE-ELM with two neural classifiers, MLP and ELM. Those two classifiers with different hidden neurons are evaluated. The fourth group of methods compared comprises fusions of three features based on an AE-ELM with the Ada-ELMs classifier. Different numbers of ELMs in the ensemble learning method of Ada-ELMs are evaluated. We then report the classification accuracy, confusion matrix, and computational time in the following sections. The implementation of ELM-related algorithms is based on High Performance toolbox for Extreme Learning Machines contributed by Akusok *et al.* [47].[2]

C. RESULTS AND ANALYSIS

We first perform experiments on the three types of features with an ELM classifier. The sigmoid function is used as the activation function. We choose a hidden layer with different numbers. Different sizes of neurons, from 256, 512, 1,024, 2,048, 4,096 to 8,192, were used. And those experiments with different settings were conducted ten times. Then, the best performances were chosen and are shown in Table 2. Evidently, the best performance in all of the settings is achieved using CNN features by the ELM classifier with 4,096 hidden neurons among all three features. The accuracy achieved in the test set is 80.6%. When using HOG as the features, the best performance is achieved by the ELM classifier with 8,192 hidden neurons. The accuracy achieved in the test set is 71.8%. When using color histograms as the features, the best performance is achieved by the ELM classifier with 4,096 hidden neurons. The accuracy achieved in the test set is 41.6%. In addition, we note that for all of those three features the accuracy increases with the number of neurons in the ELM classifier.

[2]https://pypi.python.org/pypi/hpelm

TABLE 2. Results (%) on a single feature with an ELM classifier.

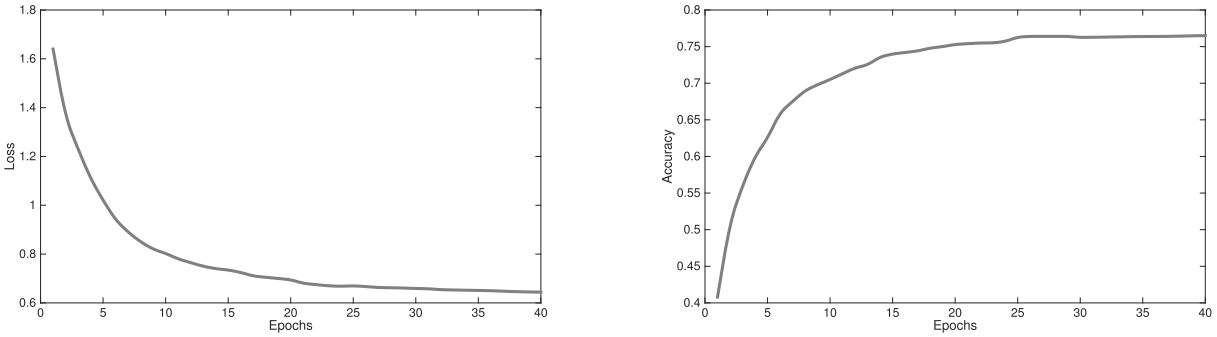
Features	256	512	1024	2048	4096	8192
CNN	78.5	79.1	79.8	80.1	80.6	80.5
HOG	62.6	64.5	66.7	68.7	70.4	71.8
Hist	39.8	40.5	41.0	41.5	41.6	41.0

Next, we perform experiments on the three types of features, taking one at a time, with an MLP classifier. The Rectified Linear Unit (ReLU) is adopted as the activation function. We set the learning rate to 0.01, the momentum to 0.95, the decay of the learning rate to 0.00018, and the batch size to 50. Besides, the momentum method is set as nesterov. We train the MLP classifier with 50 epochs. The results are shown in Table 3. Evidently, the best performance in all of the settings is achieved using CNN features, compared with the ELM classifier, by the MLP classifier with 2,048 hidden neurons. The accuracy achieved in the test set is 80.8% using 2,048 hidden neurons. When using the HOG features, the best performance is achieved by the MLP classifier with 512 hidden neurons. The accuracy achieved in the test set is 73.8% using 512 hidden neurons. When using a color histogram as the features, the best performance is achieved by the MLP classifier with 1,024 hidden neurons. The accuracy achieved in the test set is 43.5% using 1,024 hidden neurons. To summarize, using the MLP, the best performance is achieved with an accuracy of 80.8%. In contrast, using the ELM classifier, the best performance is achieved with an accuracy of 80.6%. Slightly better performance is achieved using an MLP classifier compared with an ELM classifier. Comparatively, the number of hidden neurons affects the performance of an MLP classifier more slightly than that of an ELM classifier.

TABLE 3. Results (%) on a single feature with an MLP classifier.

Features	256	512	1024	2048
CNN	80.5	80.6	80.8	80.8
HOG	73.3	73.8	72.7	73.6
Hist	42.8	43.2	43.5	43.0

Moreover, we combine those features for clothing classification. The experimental results are shown in Table 4. Evidently, the fusion of two of those feature types achieves better performance than that of using a single type of features alone. The best accuracy in the test set is 81.8% with



(a) (b)

FIGURE 3. Training Procedure of the CNN Model on the DeepFashion Dataset. (a) Training Loss Changes with Epochs. (b) Training Accuracy Changes with Epochs.

TABLE 4. Results (%) using Naive fusion with an MLP classifier.

Features	256	512	1024	2048
CNN+HOG	81.8	81.6	81.0	81.3
HOG+Hist	74.3	74.9	75.5	75.3
CNN+Hist	80.4	80.6	80.5	80.4
TRIPLE	81.4	81.3	81.0	79.5

TABLE 5. The AE-ELM reconstruction errors.

Features	256	512	1024	2048
CNN+HOG	0.22	0.14	0.08	0.05
HOG+Hist	0.26	0.16	0.09	0.06
CNN+Hist	0.16	0.11	0.07	0.04
TRIPLE	0.29	0.19	0.10	0.08

TABLE 6. Results (%) using an AE-ELM with an MLP classifier.

Features	256	512	1024	2048
CNN+HOG	80.9	81.2	81.6	82.0
HOG+Hist	70.8	72.3	73.9	74.4
CNN+Hist	80.3	80.3	80.6	80.4
TRIPLE	81.1	80.9	81.3	81.6

256 hidden neurons. And with an increment in the number of hidden units, the accuracy on the test set slightly decreases. This findings differs from those using a single feature with an ELM or an MLP classifier shown as above. To conclude, by combining different features, especially CNN and HOG, the accuracy achieved on the test set, compared with using the CNN features alone increases from 80.8% with 2,048 hidden neurons to 81.8% with 256 hidden neurons. Note that the fusion of three features, denoted with the triple in Table 4, does not shows improvement in the results. The fusion of simple concatenation does not work, which is most likely because those features belong to different representation spaces and will be deteriorated by simple concatenation.

In previous experiments, we performed fusion on two of those three features by simple concatenation. Next, we consider making deep fusion with the AE-ELM. The aim of the AE-ELM is to attempt to capture the intrinsic nature of the inputs with a compressed random projection by setting the output to the same as the input. The reconstruction errors of the AE-ELMs with different neurons are shown in Table 5. The error is averaged over the number of samples and the dimensionality of the features. Through the results shown in this table, we set the threshold to 0.10 and choose the AE-ELM with 1,024 hidden neurons.

Next, using the features by random projection with AE-ELM, MLP classifiers with different neurons are trained with 50 training epochs. The experimental results are shown in Table 6. Evidently, the fusion using AE-ELM based on the CNN and HOG features and the MLP classifier with 2,048 hidden neurons achieves the best performance. The best accuracies achieved in the test set is 82.0% with CNN and HOG features. For the fusion with CNN and Hist features, the best accuracies in the test set is 74.4%. And for the fusion with the CNN and Hist features, the best accuracy achieved is 80.6%. Compared with naive fusion, The accuracy increases from 80.8% with the CNN features alone to 82.0% with AE-ELM based fusion methods. Those findings show that by capturing the intrinsic representations, the fusion using the AE-ELM model with the random projection increases the generalization capability.

Similarly, we perform experiments using AE-ELM fusion and an ELM classifier. The results with different neurons are reported in Table 7. The best accuracies achieved in the test set is 80.7% with CNN and HOG features. However, this accuracy is lower than that using AE-ELM with an MLP classifier. As previously shown in Table 6, the results of the fusion of CNN and HOG features outperform those of using the CNN or HOG features alone, separately. Additionally, the results of the fusion of the HOG and Hist features outperform those of using the HOG or Hist features alone, separately.

Next, we show the performance of ensemble learning AdaELMs combined with those features. The number of ELMs adopted in ensemble learning is evaluated. First, we run

TABLE 7. Results (%) using an AE-ELM with an ELM classifier.

Features	256	512	1024	2048	4096	8192
CNN+HOG	77.0	78.4	78.9	79.7	80.3	80.7
HOG+Hist	62.4	67.8	69.9	71.6	72.2	72.6
CNN+Hist	77.8	78.3	79.2	79.7	79.9	80.1
TRIPLE	76.2	77.7	78.8	79.4	79.9	80.3

the CNN features and Ada-ELMs with different numbers of primitives, i.e., base classifiers. Additionally, the number of neurons is set to be from 512 to 4,096. The experimental results are reported in Table 8. The best accuracy achieved is 80.9%. Evidently, the performance is not satisfactory. We further evaluate the ensemble classifier Ada-ELMs, using AE-ELM with fusion of CNN and HOG as the features. Those results are reported in Table 9. Compared with the previous experiments, the performance is slightly improved using the ensemble learning, Ada-ELMs.

TABLE 8. Performance with Ada-ELMs using the CNN feature.

# Neurons	$k = 5$	$k = 10$	$k = 20$
512	79.8	80.0	80.0
1,024	80.1	80.3	80.3
2,048	80.6	80.6	80.6
4,096	80.7	80.8	80.9

TABLE 9. Performance with Ada-ELMs using the CNN and HOG features.

# Neurons	$k = 5$	$k = 10$	$k = 20$
512	79.7	79.9	80.0
1,024	80.1	80.3	80.3
2,048	80.5	80.8	80.7
4,096	80.9	81.1	81.0

To further investigate the performance of the Ada-ELMs, the confusion matrix for those two methods is also given in Figure 4 and Figure 5, respectively. In those two figures, the denser of the block is, the higher score of that has. We observe that the Dress class is relatively easy to classify and obtains the highest score. The Skirt class is the hardest for recognition and has the lowest score. The Skirt class is mostly above 30%, with misclassifications for Sweater and Tank. Identically, we observe that Jeans and Joggers are mostly misclassified as Shorts and Dress. In general, those two methods have some similar bottlenecks.

D. COMPUTATIONAL TIME

Finally, we show the training time that is required for those methods. Our experiments are conducted on a Dell R720 server with a dual Intel E5-2650 CPU @ 2.00 GHz and 128 GB Memory. In our framework, the time consumption is mainly due to the following modules: AE-ELM and Ada-ELMs. We record the time required for training those two modules on the DeepFashion dataset. As a comparison,

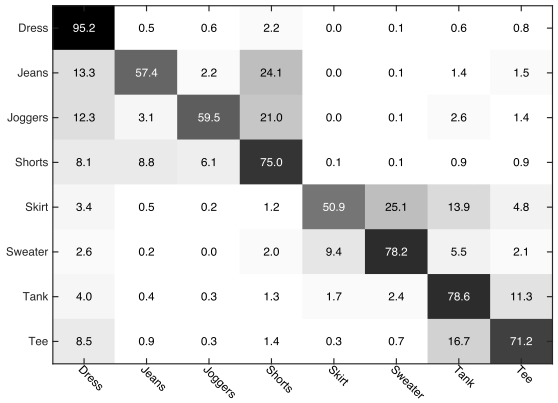


FIGURE 4. The Confusion Matrix Using the CNN Feature with Ada-ELMs.

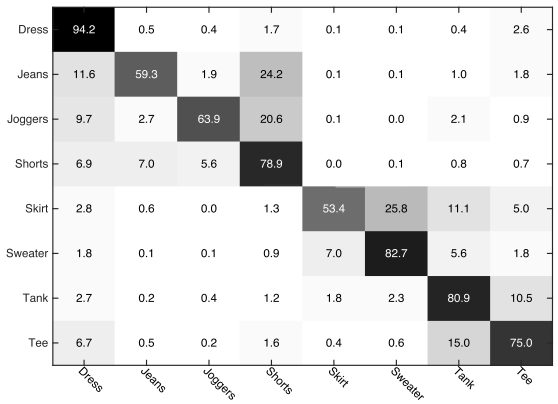


FIGURE 5. The Confusion Matrix Using Triple Features with Ada-ELMs.

TABLE 10. The typical training time (sec.) comparison.

Method	256	512	1024
MLP	1,037	1,811	2,289
ELM	2	2	5
AE-ELM	2	4	10
Ada-ELM	23	56	238

the time for training two classifiers, MLP and ELM, is also recorded. The time consumption is shown in Table 10. Specifically, if we use the MLP method for classification, we require 2,289 seconds on average for training with 1,024 hidden units. For the ELM classifier, we need only less than ten seconds. Usually, for an AE-ELM with 1,024 hidden units, we also have less than ten seconds. For the Ada-ELMs with ten primitive ELMs and in which each has 1,024 hidden units, we use approximately 238 seconds. This time consumption with Ada-ELM is one tenth that of an MLP. Notably, if we use only the AE-ELM and ELM classifier, we consume much less time compared to the other methods. For the ensemble classifier Ada-ELMs, the time required increases compared with the single ELM classifier. Even though the time increases with the number of primitive ELMs, the Ada-ELMs with ten ELMs is sufficient. The time used in those ELM-based methods is much less than for the MLP method.

V. CONCLUSIONS

In this paper, we address the task of recognizing large-scale clothing images based on multiple features and variants of ELMs. Considering the accuracy and training time, we develop a clothing recognition framework. This framework is composed of three components: low-level feature extraction, feature fusion with AE-ELM for the high-level, and ensemble classification with ELMs, i.e., Ada-ELMs. Experiments on a large-scale publicly available dataset on clothing images demonstrate that our proposed framework is flexible and competitive, especially for balancing the time and recognition accuracy.

There are still several interesting problems to be investigated in the future. For example, recognizing clothing images with imbalanced categories using those ELM-related methods is highly interesting. In this study, we consider clothing images with eight categories. However, the categories in real-world clothing images would be more than fifty. Furthermore, in recognizing fine-grained clothing images for example, a suit with different collars is valuable in practical applications. Therefore, another challenging problem is to recognize clothing images with large and fine-grained categories. In addition, exploring other types of ELMs, e.g., with sparsity, will be considered in our future research.

ACKNOWLEDGMENT

The authors would also like to thank the editor and anonymous reviewers for their invaluable comments and suggestions that allowed them to improve the final version of this article.

REFERENCES

[1] H. Chen, A. Gallagher, and B. Girod, "Describing clothing by semantic attributes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Firenze, Italy, 2012, pp. 609–623. [Online]. Available: https://doi.org/10.1007/978-3-642-33712-3_44

[2] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1096–1104. [Online]. Available: https://doi.org/10.1109/CVPR.2016.124

[3] S. Yan, Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Unconstrained fashion landmark detection via hierarchical recurrent transformer networks," in *Proc. ACM Conf. Multimedia*, 2017, pp. 172–180. [Online]. Available: https://doi.org/10.1145/3123266.3123276

[4] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool, "Apparel classification with style," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, in Lecture Notes in Computer Science, vol. 7727, 2013, pp. 321–335. [Online]. Available: https://doi.org/10.1007/978-3-642-37447-0_25

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, no. 1, Jun. 2005, pp. 886–893. [Online]. Available: https://doi.org/10.1109/CVPR.2005.177

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. New York, NY, USA: Curran Associates Inc., 2012, pp. 1097–1105.

[8] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015. [Online]. Available: https://doi.org/10.1016/j.neunet.2014.10.001

[9] O. Russakovsky and L. Fei-Fei, "Attribute learning in large-scale datasets," in *Proc. Int. Workshop Parts Attributes Eur. Conf. Comput. Vis. (ECCV)*, Crete, Greece, Sep. 2010, pp. 1–14. [Online]. Available: https://doi.org/10.1007/978-3-642-35749-7_1

[10] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable visual attributes for face verification and image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1962–1977, Oct. 2011. [Online]. Available: https://doi.org/10.1109/TPAMI.2011.48

[11] T. L. Berg, A. C. Berg, and J. Shih, "Automatic attribute discovery and characterization from noisy Web data," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 663–676. [Online]. Available: https://doi.org/10.1007/978-3-642-15549-9_48

[12] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan, "Style finder: Fine-grained clothing style detection and retrieval," in *Proc. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 8–13. [Online]. Available: https://doi.org/10.1109/CVPRW.2013.6

[13] S. Shankar, V. K. Garg, and R. Cipolla, "Deep-carving: Discovering visual attributes by carving deep neural nets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3403–3412. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298962

[14] J. Huang, R. Feris, Q. Chen, and S. Yan, "Cross-domain image retrieval with a dual attribute-aware ranking network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1062–1070. [Online]. Available: https://doi.org/10.1109/ICCV.2015.127

[15] K. Lin, H.-F. Yang, K.-H. Liu, J.-H. Hsiao, and C.-S. Chen, "Rapid clothing retrieval via deep learning of binary codes and hierarchical search," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2015, pp. 499–502. [Online]. Available: https://doi.org/10.1145/2671188.2749318

[16] Q. Dong, S. Gong, and X. Zhu, "Multi-task curriculum transfer deep learning of clothing attributes," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 520–529. [Online]. Available: https://doi.org/10.1109/WACV.2017.64

[17] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3330–3337. [Online]. Available: https://doi.org/10.1109/CVPR.2012.6248071

[18] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Parsing clothing in fashion photographs," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3570–3577. [Online]. Available: http://doi.org/10.1109/CVPR.2012.6248101

[19] Y. Kalantidis, L. Kennedy, and L.-J. Li, "Getting the look: Clothing recognition and segmentation for automatic product suggestions in everyday photos," in *Proc. 3rd ACM Conf. Int. Conf. Multimedia Retr. (ICMR)*, New York, NY, USA, 2013, pp. 105–112. [Online]. Available: https://doi.org/10.1145/2461466.2461485

[20] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Retrieving similar styles to parse clothing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 1028–1040, May 2015. [Online]. Available: https://doi.org/10.1109/TPAMI.2014.2353624

[21] P. Tangseng, Z. Wu, and K. Yamaguchi. (2017). "Looking at outfit to parse clothing." [Online]. Available: https://arxiv.org/abs/1703.01386

[22] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013. [Online]. Available: https://doi.org/10.1109/TPAMI.2013.50

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[24] K. Simonyan and A. Zisserman. (2015). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[26] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018. [Online]. Available: https://arxiv.org/abs/1710.07035

[27] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989. [Online]. Available: https://doi.org/10.1162/neco.1989.1.4.541

[28] S. Li, Z.-Q. Liu, and A. B. Chan, "Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network," *Int. J. Comput. Vis.*, vol. 113, no. 1, pp. 19–36, 2015. [Online]. Available: https://doi.org/10.1007/s11263-014-0767-8

[29] F. Feng, R. Li, and X. Wang, "Deep correspondence restricted Boltzmann machine for cross-modal retrieval," *Neurocomputing*, vol. 154, pp. 50–60, 2015. [Online]. Available: https://doi.org/10.1016/j.neucom.2014.12.020

[30] C. Lynch, K. Aryafar, and J. Attenberg, "Images don't lie: Transferring deep visual semantic features to large-scale multimodal learning to rank," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 541–548. [Online]. Available: https://doi.org/10.1145/2939672.2939728

[31] Y. Cheng, X. Zhao, R. Cai, Z. Li, K. Huang, and Y. Rui, "Semi-supervised multimodal deep learning for RGB-D object recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 3345–3351. [Online]. Available: http://www.ijcai.org/Proceedings/16/Papers/473.pdf

[32] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixel-level domain transfer," in *Proc. Eur. Conf. Comput. Vis.*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 517–532. [Online]. Available: https://doi.org/10.1007/978-3-319-46484-8_31

[33] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016. [Online]. Available: https://doi.org/10.1109/TPAMI.2015.2437384

[34] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. D. Bourdev, "PANDA: Pose aligned networks for deep attribute modeling," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 1637–1644. [Online]. Available: https://doi.org/10.1109/CVPR.2014.212

[35] Y. Bai, K. Yang, W. Yu, W.-Y. Ma, and T. Zhao, "Learning high-level image representation for image retrieval via multi-task DNN using clickthrough data," *CoRR*, vol. abs/1312.4740, 2013. [Online]. Available: http://arxiv.org/abs/1312.4740

[36] D. Wang, X. Gao, X. Wang, L. He, and B. Yuan, "Multimodal discriminative binary embedding for large-scale cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4540–4554, Oct. 2016. [Online]. Available: https://doi.org/10.1109/TIP.2016.2592800

[37] E. Simo-Serra and H. Ishikawa, "Fashion style in 128 floats: Joint ranking and classification using weak data for feature extraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 298–307. [Online]. Available: https://doi.org/10.1109/CVPR.2016.39

[38] R. Li, F. Feng, I. Ahmad, and X. Wang, "Retrieving real world clothing images via multi-weight deep convolutional neural networks," in *Cluster Computing*. New York, NY, USA: Springer, Jul. 2017. [Online]. Available: https://doi.org/10.1007/s10586-017-1052-8

[39] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012. [Online]. Available: https://doi.org/10.1109/TSMCB.2011.2168604

[40] G.-B. Huang, "What are extreme learning machines? Filling the gap between frank rosenblatt's dream and John von Neumann's puzzle," *Cogn. Comput.*, vol. 7, no. 3, pp. 263–278, 2015. [Online]. Available: https://doi.org/10.1007/s12559-015-9333-0

[41] H. Liu, L. Yu, W. Wang, and F. Sun, "Extreme learning machine for time sequence classification," *Neurocomputing*, vol. 174, pp. 322–330, Jan. 2016. [Online]. Available: https://doi.org/10.1016/j.neucom.2015.01.093

[42] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016. [Online]. Available: https://doi.org/10.1109/TNNLS.2015.2424995

[43] C. L. Lekamalage *et al.*, "Dimension reduction with extreme learning machine," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3906–3918, Aug. 2016. [Online]. Available: https://doi.org/10.1109/TIP.2016.2570569

[44] Z. Huang, Y. Yu, J. Gu, and H. Liu, "An efficient method for traffic sign recognition based on extreme learning machine," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 920–933, Apr. 2017. [Online]. Available: https://doi.org/10.1109/TCYB.2016.2533424

[45] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

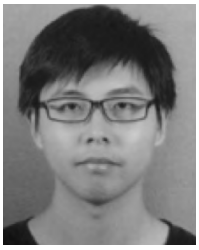
[46] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with elms for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Jun. 2013. [Online]. Available: https://doi.org/10.1109/MIS.2013.140

[47] A. Akusok, K.-M. Björk, Y. Miche, and A. Lendasse, "High-performance extreme learning machines: A complete toolbox for big data applications," *IEEE Access*, vol. 3, pp. 1011–1025, 2015. [Online]. Available: https://doi.org/10.1109/ACCESS.2015.2450498

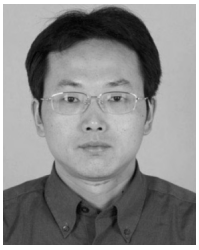
RUIFAN LI (M'13) received the B.S. and M.S. degrees in control systems, and in circuits and systems from Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001, respectively, the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2006. Since 2006, he joined the School of Computer Science, BUPT. In 2011, he spent one year as a Visiting Scholar with the Information Sciences Institute, University of Southern California, Los Angeles, CA, USA. He is currently an Assistant Professor with the School of Computer Science, BUPT, and affiliated with the Engineering Research Center of Information Networks, Ministry of Education. His current research activities include multimedia information processing, neural information processing, and statistical machine learning. Dr. Li is a member of the China Computer Federation and the Chinese Association of Artificial Intelligence. He served as an Active Reviewer for IEEE TCYB, TII, TMM, TSMC, and *Access*.

WENCONG LU received the B.E. degree from the Beijing University of Posts and Telecommunications, China, in 2017. His research interests include multimedia information processing and machine leaning.

HAOYU LIANG received the B.E. degree from the South China University of Technology, China, in 2017. He is currently pursuing the master's degree with the School of Computer Science, Beijing University of Posts and Telecommunications. His research interests include multimedia information processing and machine leaning.



YUZHAO MAO received the B.E. degree from Nanchang University, China, in 2010. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. His research interests include image caption generation and multi-modal representation learning.



XIAOJIE WANG received the Ph.D. degree from Beihang University in 1996. He is currently a Professor and the Director of the Centre for Intelligence Science and Technology, Beijing University of Posts and Telecommunications. His research interests include natural language processing and multi-modal cognitive computing. He is an Executive Member of the Council of Chinese Association of Artificial Intelligence, and the Director of the Natural Language Processing Committee. He is a member of the Council of Chinese Information Processing Society and the Chinese Processing Committee of China Computer Federation.