# Consensus Style Centralizing Auto-Encoder for Weak Style Classification

**Shuhui Jiang**[†], **Ming Shao**[†], **Chengcheng Jia**[†] and **Yun Fu**[†‡]
[†]Department of Electrical & Computer Engineering, Northeastern University, Boston, USA
[‡]College of Computer & Information Science, Northeastern University, Boston, USA
{shjiang, mingshao, cjia, yunfu}@ece.neu.edu

## Abstract

Style classification (e.g., architectural, music, fashion) attracts an increasing attention in both research and industrial fields. Most existing works focused on low-level visual features composition for style representation. However, little effort has been devoted to automatic mid-level or high-level style features learning by reorganizing low-level descriptors. Moreover, styles are usually spread out and not easy to differentiate from one to another. In this paper, we call these less representative images as **weak style** images. To address these issues, we propose a consensus style centralizing auto-encoder (CSCAE) to extract robust style features to facilitate weak style classification. CSCAE is the ensemble of several style centralizing auto-encoders (SCAEs) with consensus constraint. Each SCAE centralizes each feature of certain category in a progressive way. We apply our method in fashion style classification and manga style classification as two example applications. In addition, we collect a new dataset, Online Shopping, for fashion style classification evaluation, which will be publicly available for vision based fashion style research. Experiments demonstrate the effectiveness of SCAE and CSCAE on both public and newly collected datasets when compared with the most recent state-of-the-art works.

## Introduction

Recently, researchers have shown great interest in style classification, such as architectural style (Goel, Juneja, and Jawahar 2012; Xu et al. 2014), music style (Herlands et al. 2014), photographic style (Van Gemert 2011), manga style (Chu and Chao 2014) and fashion style (Kiapour et al. 2014; Bossard et al. 2013; Yamaguchi, Kiapour, and Berg 2013; Chao et al. 2009). For example, in vision community, fashion style, which serves as the expressions of individual's characters and aesthetics, is related to clothing parsing (Yamaguchi et al. 2012), recommendation (Wang et al. 2015; Czapiewski et al. 2015; Liu et al. 2012a) and classification (Liu et al. 2012b; Song et al. 2011; Shao, Li, and Fu 2013; Di et al. 2013), but essentially different. A particular clothing type is generally made up of a diverse set of fashion styles. For example, "suit" can be considered as both *elegant* and *renascent* fashion styles. In addition, in online shopping systems, customers could choose a special category of clothing according to personal preference or occasion, namely, fashion style. Understanding fashion styles and clothing categories are equally important. Therefore, learning robust style representation for style classification becomes an interesting research topic.



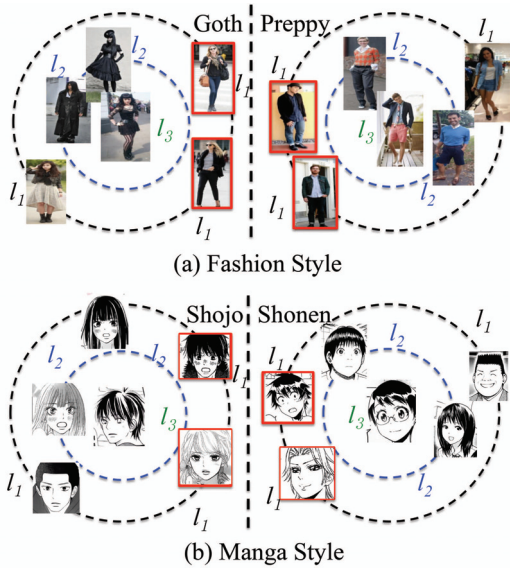


Figure 1: Illustration of "weak style" phenomenon. Style images are usually "spread out". Images in the center are representatives style images and defined as "strong style". They are easy to distinguished from other styles. We rate them at higher style level 3 (e.g., $l_3$). Images far from the center are less similar to strong style images. Images in red frames on the boundary are from two different classes, but they seem visually similar, and easily get misclassified. For example, in manga style illustration, it is hard for human to distinguish two upper mangas in the red frames into the Shojo (girl) style and Shonen (boy) style. Therefore, they are defined as "weak style" and rated at lower style level 1 (e.g., $l_1$).

Most existing style classification methods focused on extracting discriminative local patches or patterns. Goel et al. mined characteristic features with semantic utility from low-level features for different architectural styles (Goel, Juneja,

and Jawahar 2012). These characteristic features were of various scales, and provided an insight into what makes a particular architectural style category distinct. Chu et al. designed six computational features such as line orientation and angle between lines for manga style classification (Chu and Chao 2014). These features could discriminate mangas which target at young boys and young girls, and discriminated artworks produced by different artists. Recently, Kiapour et al. (Kiapour et al. 2014) released an online game to collect a fashion dataset and then proposed a style descriptor by concatenating low-level features to represent fashion styles.

However, style classification usually rely on high-level abstract concepts, and therefore style images from the same category are usually spread out. As shown in Figure 1, representative images in the center are assigned strong style level $l_3$, while less representative images distant to the center are assigned lower style level. The latter easily gets misclassified with other categories. We name them as **weak style** images and rate them to weak style level $l_1$. Furthermore, existing methods usually concatenated all the features together, meaning all the features are treated equally important. Actually, those features should be treated differently as they are considered in different fashion styles. However, to the best of our knowledge, no algorithm has been proposed to weight multiple weak style features in an automatic way.

To address above issues, we propose a consensus style centralizing auto-encoder (CSCAE) to extract robust style features, especially for weak style images. As shown in Figure 2, CSCAE is the ensemble of several style centralizing auto-encoders (SCAEs) with consensus constraint. Each SCAE centralizes the images from the same category in a progressive way using one feature. Although conventional auto-encoder (AE) could learn mid- or high-level features for style classification, the nature of weak style of style images (spread out and not easy to differentiate from others) makes style centralization necessary. To that end, we progressively transform the original input to images with higher style level using an improved AE. Conventional AE takes identical input and output, while in SCAE, for each AE building block, the corresponding output data distinction degree is one level higher than the input data. We only "pull" neighbor samples together towards the center of this category. This progressive evolution of the input data allows to slowly mitigate the weak style distinction, and ensures the smoothness of the model. For each step, our CSCAE jointly trains all the columns together with consensus constraints that keep patch weights over different feature channels consistent.

We evaluate our methods on two applications: fashion style classification and manga style classification. We also collect another new Online Shopping dataset and compare our method with the most recent state-of-the-art works on it. On both our newly collected and public datasets, our methods achieve appealing results.

The novelties of our paper could be concluded as follows:

- To the best of our knowledge, this is the first time that weak style classification problem has been identified in AI and vision community as a feature learning problem.
- We propose a new deep learning framework towards automatic mid- and high-level style feature extraction.
- We propose a consensus style centralizing auto-encoder (CSCAE) with patch weights consensus constraints on different feature channels to learn robust style features.

## Methods

In this section, we first briefly introduce the auto-encoder (AE) and denoising auto-encoder (DAE). Then we present our style centralizing auto-encoder (SCAE) and extend it to consensus style centralizing auto-encoder (CSCAE).

### Preliminary

Deep structures have been exploited to learn discriminative feature representation (Bengio 2009; Ding, Shao, and Fu 2015). Suppose $X \in \mathbb{R}^{D\times N}$ are $N$ images from the style dataset, where $D$ is the feature dimension, and $x_i \in \mathbb{R}^D$ denotes the feature of the $i$-th image in $X$. Conventional AE auto-encoder (AE) (Bengio 2009) includes two parts: encoder and decoder. An encoder attempts to map the input to the hidden layer by a linear transform and a successive non-linear activation function. Denoting the mapping as $f(\cdot)$, for each $x_i$, we can explicitly formulate this process as:

$$z_i = f(x_i) = \sigma(W_1 \times x_i + b_1), \tag{1}$$

where $z_i \in \mathbb{R}^d$ is the hidden layer representation, $W_1 \in \mathbb{R}^{d\times D}$ is the linear transform, $b_1 \in \mathbb{R}^d$ is the bias, and $\sigma$ is the non-linear activation function. In our formulation, we use the sigmoid function with the form: $\sigma(x_i) = \frac{1}{1+\exp\{-x_i\}}$.

On the other hand, the decoder $g(\cdot)$ manages to map the hidden representation $z_i$ back to the input signal $x_i$, namely,

$$x_i = g(z_i) = \sigma(W_2 \times z_i + b_2), \tag{2}$$

where the linear transform $W_2 \in \mathbb{R}^{d\times D}$ and basis $b_2 \in \mathbb{R}^D$.

To optimize the model parameters $W_1$, $b_1$, $W_2$ and $b_2$, we employ the typical least square error as the cost function:

$$\min_{\substack{W_1,b_1\\W_2,b_2}} \frac{1}{2N}\sum_{i=1}^{N}\left\|x_i - g(f(x_i))\right\|^2 + \lambda R(W_1, W_2), \tag{3}$$

where $R(W_1, W_2) = (\|W_1\|^2 + \|W_2\|^2)$ works as a regularizer and $\lambda$ is the weight decay parameter to suppress arbitrary large weights. Such problem has been widely discussed in neural networks, and can be solved by back propagation algorithms (Rumelhart, Hinton, and Williams 1988).

An effective variation of AE described above is the denoising auto-encoder (DAE) (Vincent et al. 2008). The input of DAE is the corrupted version $\tilde{X}$ by corruption process $q(\tilde{X}|X)$ of the clean data $X$. The output of DAE is the clean data $X$. During the training processing, DAE learns a stochastic mapping $p(X|\tilde{X})$ (Vincent et al. 2010), which reconstructs the corrupted $\tilde{X}$ back to the uncorrupted ones. Mathematically, the cost function can be written as:

$$\min_{\substack{W_1,b_1\\W_2,b_2}} \frac{1}{2N}\sum_{i=1}^{N}\left\|x_i - g(f(\tilde{x}_i))\right\|^2 + \lambda R(W_1, W_2). \tag{4}$$

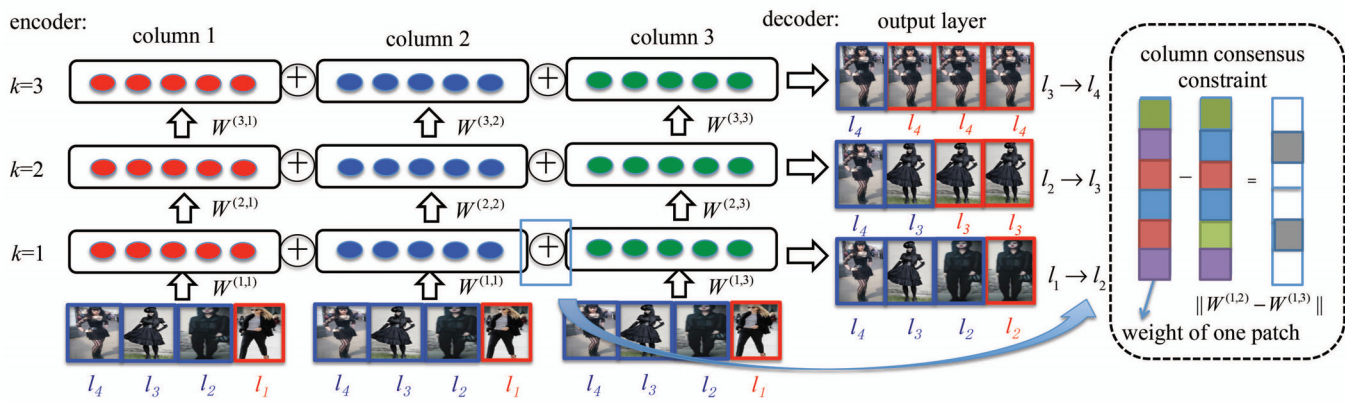


Figure 2: Consensus style centralizing auto-encoder (CSCAE) framework. Each **column** (red, blue, green) represents a style centralizing auto-encoder (SCAE). Example images of each level $l$ are presented with colored frames. In step $k$, samples in $l_k$ are replaced by the nearest neighbors in $l_{k+1}$ (red). Samples in higher level than $l_k$ are not changed (blue). In CSCAE, different feature channels (columns) are trained jointly by adding column consensus constraint on each patch.

## Style Centralizing Auto-Encoder (SCAE)

In this section, we introduce style centralizing auto-encoder (SCAE) in detail, and explain how SCAE centralizes weak style to strong style from the manifold learning perspective.

Conventional AE takes identical input and output, while in SCAE, the corresponding output data's style level is one level higher than the input data. This progressive evolution of the input data allows to slowly mitigate the weak style distinction, and ensures the smoothness of the model. This progressive thought is somehow similar to the design of DAE (Vincent et al. 2008); however SCAE explicitly carries the semantics of weak style, by which the weak style features could be projected back to the strong style features.

Illustration of full pipeline of CSCAE can be found in Figure 2, in which each column is a SCAE for a feature channel. In each step $k$, the input $X_k$ can be seen as the "corrupted" version of $X_{k+1}$ by the nearest neighbor rule found from samples in $l_k$. Samples in higher level than $l_k$ are not changed (blue). To clarify this, we take the first layer of SCAE for example. The inputs of the 1st layer of SCAE are the features of images in ascent order w.r.t style level, namely, $X_1$, $X_2$, $X_3$ and $X_4$. The corresponding output of $X_1$ is $X_2$, which is one level higher. We use the identical input and output for images in $l_2$, $l_3$ and $l_4$ in the 1st layer of SCAE. For the inputs of the 2nd layer, as images in $l_1$ are already transformed to $l_2$, we treat them as $l_2$ images in the second layer. Mathematically, SCAE can be formulated as: suppose we have $L$ style levels, the $k$-th layer of SCAE for category $c$ can be written as:

$$\min_{\substack{W_{1,k}^{(c)},b_{1,k}^{(c)} \\ W_{2,k}^{(c)},b_{2,k}^{(c)}}} \sum_{i,j \quad x_{j,k}^{(c)} \in u(x_{i,k+1}^{(c)})} \left\| x_{i,k+1}^{(c)} - g(f(x_{j,k}^{(c)})) \right\|^2 + \sum_{\xi=k+1}^{L} \sum_{i} \left\| x_{i,\xi}^{(c)} - g(f(x_{i,\xi}^{(c)})) \right\|^2 + \lambda R(W_{1,k}^{(c)}, W_{2,k}^{(c)}) \tag{5}$$

where $x_{j,k}^{(c)} \in u(x_{i,k+1}^{(c)})$ means $x_{j,k}^{(c)}$ is the nearest neighbor of $x_{i,k+1}^{(c)}$. The problem above can be solved in the same way as problem in Eq. (3). Similarly, the deep structure can be built in a layer-wise way, which is outlined in Algorithm 1.

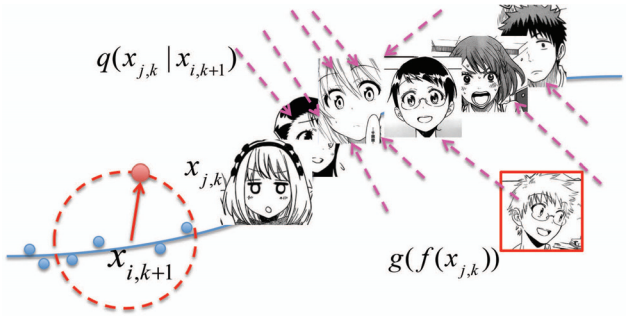


Figure 3: Manifold learning perspective of SCAE with mangas in "shonen" style. Suppose the higher level ($l_{k+1}$) mangas (or blue points) lie close to a low dimensional manifold, then samples $X_k$ in lower level ($l_k$) obtained by corruption process $q(X_k|X_{k+1})$ will lie far from the manifold. An example of a manga $x_{j,k}$ in $l_k$ is shown in the bottom right corner. The mapping operator $p(X_{k+1}|X_k)$ manages to project $x_{j,k}$ back to onto the manifold through encoder $f(\cdot)$ and decoder $g(\cdot)$ processing.

**Geometric Interpretation of SCAE**: The process of centralizing which maps a weak style sample back to a strong style can be interpreted from the geometric perspective under the manifold assumption (Chapelle et al. 2006). Vincent et al. provided the manifold learning perspective regarded to the insight behind the DAE (Vincent et al. 2010). Suppose the uncorrupted data lie close to a non-linear manifold. Usually, the non-linear manifold is with lower dimension. The corrupted version $\tilde{x}$ of $x$ is obtained by a stochastic mapping $q(\tilde{X}|X)$. During the training, DAE learns a stochastic operator $p(X|\tilde{X})$ which maps the corrupted data $\tilde{x}$ back to $x$ which is on or close to the manifold.

Figure 3 illustrates the manifold learning perspective of SCAE where images of shonen style mangas are shown as the examples. Suppose the higher level ($l_{k+1}$) mangas lie close to a low dimensional manifold. The weak style examples are more likely being far away from the manifold than the higher level ones. Note $x_{j,k}$ is the corrupted version of $x_{i,k+1}$ by the operator $q(X_k|X_{k+1})$, and therefore lies far away from the manifold. In SCAE, $q(X_k|X_{k+1})$ manages to find the nearest neighbor of $x^{(c)}_{i,k+1}$ in level $l_k$ to obtain the corrupted the vision of $x^{(c)}_{i,k+1}$ as $x^{(c)}_{j,k+1}$. During the centralizing training, similar to DAE, SCAE learns the stochastic operator $p(X_{k+1}|X_k)$ that maps the lower style level samples $X_k$ back to a higher level. Successful centralization implies that the operator $p$ is able to map spread-out weak level style data back to the high level style data which are close to the manifold.

## Consensus Style Centralizing Auto-Encoder (CSCAE)

Consensus style centralizing auto-encoder (CSCAE) is the ensemble of several SCAEs with column consensus constraint. As shown in Figure 2, each column (red, blue or green) is a SCAE centralizing one kind of low-level feature. As it is difficult to manually assign the weights of different low-level features (e.g., HOG, RGB), we propose to use CSCAE for automatic features weighting.

To that end, we propose to add patch weight consensus constraint through minimizing the differences of weights of the same patch from different feature channels. Each patch presents one semantic region of the image. For example, in fashion style images, a patch could be the region of arm or waist, and in mangas, it could be the region of eye or mouth. Intuitively, the weights of the patch from different feature channels should be similar, as they encode the exact same matter but in different ways. Taking face recognition as an example, the eye patch should be more important than cheek patch, as demonstrated by many face recognition works. Back to our consensus model, taking manga as an example, although the input features are different, the eye patches in different columns should be equally important.

We also illustrate the principle of column consensus in the right part of Figure 2. Each cell represents one patch and different colors are used to distinguish the patch weights. Weights from low to high are presented by colors from light to dark. Take column 2 and column 3 in step $k = 1$ as an example. $\|W^{(1,2)} - W^{(1,3)}\|$ is the difference between the weights of these two columns, where $W^{(k,\mu)}$ indicates the weight matrices in the $\mu$-th column in step $k$, and $w_i^{(k,\mu)}$ presents the $i$-th patch in $W^{(k,\mu)}$. If these two columns follow our consensus assumption, meaning either both columns assign high weights to patch $i$, or both columns set low weights to patch $i$, in either case, $\left\|w_i^{(1,2)} - w_i^{(1,3)}\right\|$ should be very small and close to zero, which is presented in a white color. Otherwise, $\left\|w_i^{(1,2)} - w_i^{(1,3)}\right\|$ should be large and presented in a dark color.

To fulfill the assumptions above, we add the KL divergence $KL(\hat{\rho}\|\rho)$ to multi-column SCAE model as a new regularizer to minimize the differences of patch weights in different columns, and obtain the new cost function $J_k^{'}$ of the $k$-th step:

$$J_k^{'} = \sum\nolimits_{\mu=1}^{N_f} J_{\mu,k} + \beta KL(\hat{\rho}\|\rho), \tag{6}$$

where $N_f$ is number of feature channels, and $\beta$ is the parameter to adjust the weight of the $KL(\hat{\rho}\|\rho)$. The left part is the sum of reconstruction cost of all the columns, where $J_{\mu,k}$ is the cost function of the $\mu$-th column in the $k$-th step, which can be calculated in the similar way as Eq. (5). The right part is the regularizer for column consensus as discussed above, where $KL(\hat{\rho}\|\rho)$ can be computed as:

$$KL(\hat{\rho}\|\rho) = \rho \log \frac{\rho}{\hat{\rho}} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}}, \tag{7}$$

where $\rho$ is a model parameter, and $\hat{\rho}$ is the average patch weight differences over different columns, calculated as:

$$\hat{\rho} = \frac{1}{N_f \times (N_f - 1)} \sum_{\mu > \nu} \left\|W^{(k,\mu)} - W^{(k,\nu)}\right\|^2, \tag{8}$$

where $W^{(k,\mu)}$ and $W^{(k,\nu)}$ are the weight matrices in the $\mu$-th column and $\nu$-th column in step $k$. The regularizer detailed in Eq. (7) essentially minimizes the difference of pre-defined $\rho$ and $\hat{\rho}$. By setting $\rho$ at a very small value, we could get a sparse structure of $\hat{\rho}$, which fulfills our assumption that the differences of weights of the same patch from different columns should be very small.

**Algorithm 1** Style Centralizing Auto-Encoder

**INPUT**: Style feature $X$ including weak style feature.
**OUTPUT**: Style centralizing feature $Z_k$, model parameters: $W^{(c)}_{1,k}$, $W^{(c)}_{2,k}$, $b^{(c)}_{1,k}$, $b^{(c)}_{2,k}$, $k \in [1, L-1]$, and $c \in \{1, ..., N_c\}$.

```
Initial Z^(0) = X.
for k=1,2,...,L-1 do
    X^(k)=Z^(k-1).
    for c = 1,..., N_c do
        Calculate W^(c)_{1,k}, W^(c)_{2,k}, b^(c)_{1,k}, b^(c)_{2,k} by Eq. (5).
        Calculate Z^(c)_k by Eq. (1).
    end for
    Combine all Z^(c)_k, c ∈ {1, ..., N_c} into Z_k.
end for
```

### Solutions

Here we describe how to solve the objective function proposed in Eq. (6). Although we still use stochastic gradient descent + back propagation for solutions, we have to jointly consider multiple columns updating corresponding to different features, which is linked by Eq. (8). In brief, the basic gradient updating rules for model parameters $W^{(k,\mu)}$, $b^{(k,\mu)}$ of the $\mu$ column, $k$-th layer of auto-encoder will be:

$$W^{(k,\mu)}_{i,j} := W^{(k,\mu)}_{i,j} - \frac{\partial}{\partial W^{(k,\mu)}_{i,j}} J_k^{'}(W, b), \tag{9}$$

$$b_i^{(k,\mu)} := b_i^{(k,\mu)} - \frac{\partial}{\partial b_i^{(k,\mu)}} J_k^{'}(W,b), \tag{10}$$

where $i$ and $j$ are the index of input and output nodes of $k$-th layer, respectively. $W_{i,j}^{(k,\mu)}$ and $b_i^{(k,\mu)}$ are the weight matrix and weight bias of mapping the $i$-th node of the input of the $k$-th layer to the $j$-th node of the output of this layer.

The key procedure is finding partial derivative $\frac{\partial}{\partial W_{i,j}^{(k,\mu)}} J_k^{'}(W,b)$ and $\frac{\partial}{\partial b_i^{(k,\mu)}} J_k^{'}(W,b)$ by the back propagation algorithm. For node $i$ in the output layer $L$, we measure the error which caused by node through calculating the difference between the true target value and the output activation of the network, and denote the error term as $\delta_i^{(L,\mu)}$. For the objective function in general AE as Eq. (3), for the hidden layer ($k < L$), the error term $\delta_i^{(k,\mu)}$ can be computed by:

$$\delta_i^{(k,\mu)} = \Big( \sum_j W_{ji}^{(k,u)} \delta_j^{(k+1,\mu)} \Big) f'(z_i^{(k,\mu)}). \tag{11}$$

In CSCAE, since the consensus constraint $KL(\hat{\rho}\|\rho)$ is added into Eq. (7), the $\delta_i^{(k,\mu)}$ can be calculated as:

$$\left( \sum_j W_{ji}^{(k,\mu)} \delta_j^{(k+1,\mu)} + \beta \Big( - \frac{\rho}{\hat{\rho}} + \frac{1-\rho}{1-\hat{\rho}} \Big) \right) f'(z_i^{(k,\mu)}). \tag{12}$$

Finally, we obtain partial derivatives as:

$$\frac{\partial}{\partial W_{i,j}^{(k,\mu)}} J_k^{'}(W,b) = a_j^{(k,\mu)} \delta_i^{(k+1,\mu)}, \tag{13}$$

$$\frac{\partial}{\partial b_i^{(k,\mu)}} J_k^{'}(W,b) = \delta_i^{(k+1,\mu)}, \tag{14}$$

where $a_j^{(k,\mu)} = z_j^{(k-1,\mu)}$ is the $j$-th input node of $\mu$-th column of the $k$-th layer.

We solve the problem above using L-BFGS optimizer (Nocedal 1980; Ngiam et al. 2011) since it can solve large-scale problems only using limited memory. As the weights of different columns are adaptively learned after minimizing the cost function, we do not manually assign weights of columns any more. Afterwards, we use hidden layer $z_i$ as the learned representation for style feature. To train the deep model in a more efficient way, we employ layer-wise training procedure. All hidden layers from the deep model are stacked to form the final style representation.

# Experiment

We evaluate our methods on two applications, fashion style classification and manga style classification. Firstly we introduce dataset processing and the state-of-art methods of these two applications. Then classification performances of our methods and the state-of-art methods are reported.

## Dataset Processing

**Hipster Wars (Kiapour et al. 2014).** It contains 5 categories and 1893 images of fashion style (Kiapour et al. 2014). It provides reliable human judgments of style level.

**Online Shopping.** It is collected by us from online shopping websites (e.g., "Nordstrom.com", "barneys.com") containing more than 30,000 images. We invited 7 professionals to manually label each image to one of 12 classes according to the category definition by fashion magazine (Chang et al. 2003). For image $i$, we calculated how many people labeled this image to category $j$, $j \in \{1, ..., 12\}$, denoted by $\phi_i^{(j)}$. We chose the $j^*$-th category, where $j^* = \arg\max_j \phi_i^{(j)}$, as the groundtruth and $\phi_i^{(j^*)}$ is regarded as style level for image $i$. Due to the space limitation, more descriptions of Online Shopping dataset will be provided in the future release.

For fashion images in both Hipster Wars dataset and Online Shopping dataset, first, pose estimation is applied to extract key boxes of human body (Yang and Ramanan 2011). Note that for Hipster Wars, we use full body bounding box, while for Online Shopping, we only use upper body bounding box since Online Shopping images are upper body centered. The bounding box of external dataset is extracted accordingly. We then extract 7 dense features for each box: RGB color value, LAB color value, HSI color value, Gabor, MR8 texture response (Varma and Zisserman 2005), HOG descriptor (Dalal and Triggs 2005), and probability of pixels which belongs to skin categories [1]. Finally, we split each box into 4 patches (2×2) and extract features with mean-std pooling.

**Manga (Chu and Chao 2014).** Chu et al. collected a shonen and shojo mangas databased including 240 panels. Six computational features: including angle between lines, line orientation, density of line segments, orientation of nearby lines, number of nearby lines with similar orientation and line strength, are calculated from each panel. We apply K-means to cluster the images based on low-level features and split them to different style levels according to their distances from the centroids.

In all the classification tasks of fashion and manga, a 9:1 training to test ratio is used for training-test process, and we repeat it for 50 times. Then SVM classifier is applied in Hipster Wars and Manga dataset by following the settings in (Kiapour et al. 2014; Chu and Chao 2014), while nearest neighbor classifier is applied on Online Shopping dataset.

## Competitive Methods

As our method focuses on fashion and manga style feature extraction, in the following part, we mainly compare with the state-of-the-art fashion/clothing style/(first three) and manga style (the fourth) feature extraction methods.

**[ECCV, 2014] (Kiapour et al. 2014)**: This method concatenated all the low-level features after mean-std pooling as the input of the classifier and named as style descriptor.

**[ICCV, 2013] (Yamaguchi, Kiapour, and Berg 2013)**: This work applied PCA after concatenating all the pooled features to reduce the dimension. Then the compressed features were taken as the input of the classifier.

**[ACCV, 2012] (Bossard et al. 2013)**: This method learned a codebook through K-means after low-level feature

[1] http://kr.mathworks.com/matlabcentral/fileexchange/28565-skin-detection

Table 1: Performances of fashion style classification of 6 methods on Hipster Wars dataset.

| Performance | $p$=0.1 | $p$=0.2 | $p$=0.3 | $p$=0.4 | $p$=0.5 |
|---|---|---|---|---|---|
| [ECCV, 2014] | 77.73 | 62.86 | 53.34 | 37.74 | 34.61 |
| [ICCV, 2013] | 75.75 | 62.42 | 50.53 | 35.36 | 33.36 |
| [ACCV, 2012] | 76.36 | 62.43 | 52.68 | 34.64 | 33.42 |
| SCAE (Ours1) | 84.37 | 72.15 | 59.47 | 48.32 | 38.41 |
| MC-SCAE (Ours2) | 87.42 | **77.00** | 62.42 | 51.68 | 41.54 |
| CSCAE (Ours3) | **89.21** | 75.32 | **64.55** | **52.88** | **43.77** |

Table 2: Performances of fashion style classification of 6 methods on Online Shopping dataset.

| Performance | $\phi$=7 | $\phi$=6 | $\phi$=5 | $\phi$=4 | $\phi$=3 |
|---|---|---|---|---|---|
| [ECCV, 2014] | 60.92 | 58.52 | 54.57 | 48.63 | 42.40 |
| [ICCV, 2013] | 55.00 | 53.96 | 51.73 | 45.38 | 37.91 |
| [ACCV, 2012] | 54.58 | 59.43 | 52.47 | 41.39 | 35.42 |
| SCAE (Ours1) | **74.33** | 61.93 | 60.72 | 50.13 | 48.89 |
| MC-SCAE (Ours2) | 66.15 | 62.53 | 62.54 | **53.28** | 49.52 |
| CSCAE (Ours3) | 70.41 | **68.42** | **63.84** | 51.18 | **50.31** |

extraction. Then the bag-of-words features were further processed by spatial pyramids and max-pooling.

**[MM, 2014] (Chu and Chao 2014)**: This method concatenated six low-level manga features as the input of SVM to classify the mangas to either shonen or shojo style.

**SCAE (Ours1)**: This method encodes the style descriptor in (Kiapour et al. 2014) into style centralizing auto-encoder (SCAE), meaning the early fusion is applied to all the low-level features before SCAE.

**MC-SCAE (Ours2)**: Multi-column SCAE. This is different from the proposed consensus style centralizing auto-encoder as it did not consider the consensus constraint in the training. Instead, it trains multiple SCAEs independently, one column at a time. Then a late fusion is applied to the encoded features according to the work in (Agostinelli, Anderson, and Lee 2013).

**CSCAE (Ours3)**: This method contains the full pipeline of the proposed consensus style centralizing auto-encoder.

### Results on Hipster Wars Dataset (Public)

Table 1 shows the accuracy (%) of competitive methods and our methods under $p = 0.1, ..., 0.5$ where $p$ determines the percentage of top ranked data according to their style level which was used the in classification task (Kiapour et al. 2014). The default settings of SCAE, MC-SCAE and CSCAE are $L = 4$, and the layer size is 400. In addition we set $\rho$=0.05, $\lambda = 10^{-5}$ and $\beta = 10^{-2}$.

First, from Table 1 we can see that results of the proposed SCAE, MC-SCAE and CSCAE are superior compared to the existing works [ECCV,2014], [ICCV, 2013] and [ACCV, 2012]. Notably, SCAE with deep structure to centralize weak style features is already better than [ECCV, 2014] by 6.64%, 9.09%, 6.13%, 10.58% and 3.80% under $p$ from 0.1 to 0.5. MC-SCAE and CSCAE outperform SCAE under all the conditions, which means multi-column strategy is able to achieve exert positive effect. CSCAE outperforms MC-SCAE under most of the conditions, which demonstrates the importance of column consensus constraints.

### Results on Online Shopping Dataset (Collected)

Table 2 shows the accuracy (%) of 6 methods under $\phi = 7, 6, 5, 4, 3$. We use nearest neighbor as the classifier and empirically set the number of neighbors as 5. Other settings are the same as those used in Table 1. From these results, we can observe that performance on the newly collected Online Shopping dataset is similar to that shown in Table 1. Note that our method CSCAE containing the whole pipeline of consensus style centralizing auto-encoder performs best. Specifically, our performance is higher than [ECCV, 2014] by 9.49%, 10.10%, 8.74%, 2.55% and 7.19% under $\phi$ from 7 to 3.

Table 3: Performances of manga style classification of 4 methods on Manga dataset.

| Performance | $p$=0.1 | $p$=0.2 | $p$=0.3 | $p$=0.4 | $p$=0.5 |
|---|---|---|---|---|---|
| [MM, 2014] | 83.21 | 71.35 | 68.62 | 64.79 | 60.07 |
| SCAE (Ours1) | 83.75 | 73.43 | 69.32 | 65.42 | 63.60 |
| MC-SCAE (Ours2) | 85.38 | 72.93 | 71.48 | **69.58** | 65.48 |
| CSCAE (Ours3) | **86.23** | **76.93** | **73.28** | 68.63 | **67.35** |

### Results on Manga Dataset (Public)

Table 3 shows the accuracy (%) of 3 proposed methods and the state-of-art method [MM,2014] under $p = 0.1, ..., 0.5$. $p$ has the same definition as experiments on Hipster Wars. From these results, we can observe that performances of our methods are better than the state-of-the-art. Specifically, our method with consensus constraint outperforms [MM,2014] by 3.02%, 5.58%, 4.26%, 3.84% and 7.28% under $p$ from 0.1 to 0.5.

## Conclusions

In this paper, we proposed a consensus style centralizing auto-encoder (CSCAE) to extract robust style feature presentation for style classification. First, SCAE progressively drew weak style images to the class center. Second, column consensus constraints automatically allocated the weights for different style features. We applied our methods on fashion style classification and manga style classification. Extensive experimental results on fashion and manga style classification demonstrated that both SCAE and CSCAE were effective for these tasks, and outperformed recent state-of-the-art works.

## Acknowledgment

This research is supported in part by the NSF CNS award 1314484, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, NPS award N00244-15-1-0041, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

## References


Agostinelli, F.; Anderson, M. R.; and Lee, H. 2013. Adaptive multi-column deep neural networks with application to robust image denoising. In *Advances in Neural Information Processing Systems*, 1493–1501.

Bengio, Y. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning* 2(1):1–127.

Bossard, L.; Dantone, M.; Leistner, C.; Wengert, C.; Quack, T.; and Van Gool, L. 2013. Apparel classification with style. In *Asian Conference on Computer Vision*. Springer. 321–335.

Chang, Y.-C.; Chuang, M.-C.; Hung, S.-H.; Shen, S.-J. C.; and Chu, B. 2003. A kansei study on the style image of fashion design. In *the 6th Asian Design Conference*.

Chao, X.; Huiskes, M. J.; Gritti, T.; and Ciuhu, C. 2009. A framework for robust feature selection for real-time fashion style recommendation. In *International workshop on Interactive multimedia for consumer electronics*, 35–42. ACM.

Chapelle, O.; Schölkopf, B.; Zien, A.; et al. 2006. Semi-supervised learning.

Chu, W.-T., and Chao, Y.-C. 2014. Line-based drawing style description for manga classification. In *ACM International Conference on Multimedia*, 781–784. ACM.

Czapiewski, P.; Forczmański, P.; Frejlichowski, D.; and Hofman, R. 2015. Clustering-based retrieval of similar outfits based on clothes visual characteristics. In *Image Processing & Communications Challenges 6*. Springer. 29–36.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 886–893. IEEE.

Di, W.; Wah, C.; Bhardwaj, A.; Piramuthu, R.; and Sundaresan, N. 2013. Style finder: Fine-grained clothing style detection and retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 8–13. IEEE.

Ding, Z.; Shao, M.; and Fu, Y. 2015. Deep low-rank coding for transfer learning. In *International Joint Conference on Artificial Intelligence*, 3453–3459.

Goel, A.; Juneja, M.; and Jawahar, C. 2012. Are buildings only instances?: exploration in architectural style categories. In *Indian Conference on Computer Vision, Graphics and Image Processing*. ACM.

Herlands, W.; Der, R.; Greenberg, Y.; and Levin, S. 2014. A machine learning approach to musically meaningful homogeneous style classification. In *AAAI Conference on Artificial Intelligence*, 276–282.

Kiapour, M. H.; Yamaguchi, K.; Berg, A. C.; and Berg, T. L. 2014. Hipster wars: Discovering elements of fashion styles. In *European Conference on Computer Vision*. Springer. 472–488.

Liu, S.; Feng, J.; Song, Z.; Zhang, T.; Lu, H.; Xu, C.; and Yan, S. 2012a. Hi, magic closet, tell me what to wear! In *ACM international conference on Multimedia*, 619–628. ACM.

Liu, S.; Song, Z.; Liu, G.; Xu, C.; Lu, H.; and Yan, S. 2012b. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3330–3337. IEEE.

Ngiam, J.; Coates, A.; Lahiri, A.; Prochnow, B.; Le, Q. V.; and Ng, A. Y. 2011. On optimization methods for deep learning. In *International Conference on Machine Learning*, 265–272.

Nocedal, J. 1980. Updating quasi-newton matrices with limited storage. *Mathematics of computation* 35(151):773–782.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5:696–699.

Shao, M.; Li, L.; and Fu, Y. 2013. What do you do? occupation recognition in a photo via social context. In *IEEE International Conference on Computer Vision*, 3631–3638. IEEE.

Song, Z.; Wang, M.; Hua, X.-s.; and Yan, S. 2011. Predicting occupation via human clothing and contexts. In *IEEE International Conference on Computer Vision*, 1084–1091. IEEE.

Van Gemert, J. C. 2011. Exploiting photographic style for category-level image classification by generalizing the spatial pyramid. In *ACM International Conference on Multimedia Retrieval*, 1–8. ACM.

Varma, M., and Zisserman, A. 2005. A statistical approach to texture classification from single images. *International Journal of Computer Vision* 62(1-2):61–81.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine learning*, 1096–1103. ACM.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research* 11:3371–3408.

Wang, H.; Zhou, Z.; Xiao, C.; and Zhang, L. 2015. Content based image search for clothing recommendations in e-commerce. In *Multimedia Data Mining and Analytics*. Springer. 253–267.

Xu, Z.; Tao, D.; Zhang, Y.; Wu, J.; and Tsoi, A. C. 2014. Architectural style classification using multinomial latent logistic regression. In *European Conference on Computer Vision*. Springer. 600–615.

Yamaguchi, K.; Kiapour, M. H.; Ortiz, L. E.; and Berg, T. L. 2012. Parsing clothing in fashion photographs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3570–3577. IEEE.

Yamaguchi, K.; Kiapour, M. H.; and Berg, T. L. 2013. Paper doll parsing: Retrieving similar styles to parse clothing items. In *IEEE International Conference on Computer Vision*, 3519–3526. IEEE.

Yang, Y., and Ramanan, D. 2011. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1385–1392. IEEE.