

Deep Fashion Recommendation System with Style Feature Decomposition

Yong-Goo Shin, Yoon-Jae Yeo, Min-Cheol Sagong, Seo-Won Ji, and Sung-Jea Ko, *Fellow*, IEEE
School of Electrical Engineering, Korea University, Seoul 136-713, Republic of Korea
 Email: sjko@korea.ac.kr

Abstract — Recently, convolutional neural network (CNN) - based fashion recommendation techniques, which automatically recommend the matching clothes to the consumer, have been widely researched. In general, the feature vector of a fashion item, i.e. clothes vector, obtained by CNN conveys two types of information: style and category, where the style indicates the distinctive characteristic of the clothes and the category represents the common properties of the clothes in the same class. Due to the mixed information of style and category, however, the clothes vector often recommends the unmatching clothes. To solve this problem, we propose a style feature extraction (SFE) layer, which effectively decomposes the clothes vector into style and category. Based on the characteristics that the category information has small variations in the same class while being distinguished from other classes, we extract and remove the category information from the clothes vector to obtain more accurate style information. Experimental results show that the proposed method achieves state-of-the-art results in terms of link prediction, which is a performance measure of a stylish match. In addition, as a simple CNN layer, it is expected that the proposed SFE layer is compatible with all popular CNN architectures.

Keywords—deep learning, convolutional neural network, recommendation system, visual compatibility

I. INTRODUCTION

In recent years, various deep learning-based shopping platforms have been launched to provide some conveniences to the user. Among those platforms, clothing recommendation systems commonly offer consumers either a whole outfit or another fashion item that matches well with certain clothes [1]. Specifically, the whole outfit recommendation system suggests a collection of clothes, which make up whole outfits with similar attributes, such as patterns and colors [2]-[6], whereas the stylish match recommendation (SMR) system models the style characteristics of clothes in order to recommend fashion items that make good matches with given clothes, e.g. a shirt and a matching pair of pants [7]-[11]. This paper focuses on the SMR system.

In order to build a feature space describing the relationship between pairs of clothes, the conventional SMR systems transform the clothes image into the feature vector called a clothes vector, using the convolutional neural networks

(CNNs). McAuley *et al.* [7] utilized the parameterized distance metric, i.e. Mahalanobis distance, to measure the distance between two clothes vectors in the feature space. He *et al.* [8] extended the work of [7] by using the k Mahalanobis transformation to obtain the multiple vector distances in the k feature spaces. However, these two methods utilize the clothes vector obtained from the CNN that is pre-trained for object classification in order to learn their distance metric. Therefore, the clothes vector does not represent enough clothing style characteristics. Viet *et al.* [9] employed the Siamese CNN to predict whether two clothing items from different categories fit together. However, the clothes vector obtained through this method is not effective in predicting compatibility between items from different categories since it contains mixed information from the category and style [12]. Recently, Liu *et al.* [12] tried to resolve this problem by subtracting the category information from a clothes vector. However, this method is limited in its ability to capture accurate style information since it could not effectively learn the category information using their objective function.

In order to obtain precise style information based on Siamese CNN, we propose a style feature extraction (SFE) layer which decomposes the clothes vector into style and category. In the SFE layer, we extract and remove the category information from the clothes vector based on the properties that the category information has small variations in the same class whereas being distinguished from other classes. To this end, our SFE layer is designed by employing a fully-connected layer, which produces category information and is trained using two different loss functions: the softmax cross-entropy loss function [13, 14], i.e. softmax loss, and the center loss function [11], which is useful to obtain features having inter-class dispersion and intra-class compactness. The empirical results using the Amazon dataset [7] exhibit that the CNNs with the proposed SFE layer accomplish state-of-the-art results in terms of link prediction, which is a performance measure of stylish matching. In addition, as a simple CNN layer, it is expected that the proposed SFE layer is compatible with all popular CNN architectures, such as GoogLeNet [13], VGG16 [14], and MobileNet [16], and is easy to implement.

II. PRELIMINARIES

A. Siamese CNN

The Siamese CNN has been commonly employed for

This work was supported by Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIP) (No. 2019-0-00268, Development of SW technology for recognition, judgement and path control algorithm verification simulation and dataset generation)

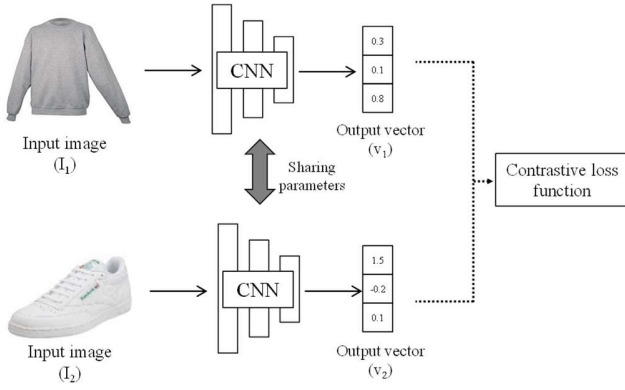


Fig. 1. The training procedure of the Siamese CNN. The Siamese CNN consists of two identical base CNNs which share their parameters, and they generate two output vectors. The Siamese CNN is trained by minimizing the contrastive loss function which compares the two output vectors.

learning a similarity metric using pairs of data labeled as matching or non-matching [17]-[19]. As shown in Fig. 1, the Siamese CNNs pass image pairs I_1 and I_2 through two identical base CNNs, such as VGG16 [14], and feed the two output vectors from the last layer of base CNNs, $\mathbf{v}_1$ and $\mathbf{v}_2$, to the contrastive loss function [17], which indicates how well the base CNN is able to place the output vectors of matching images close together and keep the output vectors of non-matching images separated in the feature space. The contrastive loss function is expressed as follows:

$$L_{\text{contrastive}}(\mathbf{v}_1, \mathbf{v}_2) = \frac{1}{2} \cdot Y \cdot \|\mathbf{v}_1 - \mathbf{v}_2\|_2^2 + \frac{1}{2} \cdot (1 - Y) \cdot \left\{ \max(0, M - \|\mathbf{v}_1 - \mathbf{v}_2\|_2) \right\}^2, \quad (1)$$

where Y represents an indicator variable which is $Y=1$ ($=0$) for a mating (non-matching) pair of I_1 and I_2 and M is a margin parameter affecting non-matching pairs.

Viet *et al.* [9] employed the Siamese CNN to predict whether clothing pairs from two different categories match well or not. However, this method often results in a prediction error since the clothes vector obtained by the Siamese CNN contains mixed information from the style and category. In the next subsection, we introduce the SFE layer which decomposes the clothes vector into style and category.

III. PROPOSED METHOD

In [12], the clothes vector $\mathbf{v}$ consists of two components style $\mathbf{v}_s$ and category $\mathbf{v}_c$ as follows:

$$\mathbf{v} = \mathbf{v}_c + \mathbf{v}_s. \quad (2)$$

We obtain $\mathbf{v}_s$ by removing $\mathbf{v}_c$ having inter-category dispersion and intra-category compactness as much as possible, which is essential to category information, from $\mathbf{v}$. As depicted in Fig. 2, the proposed SFE layer produces $\mathbf{v}_c$ using a fully connected

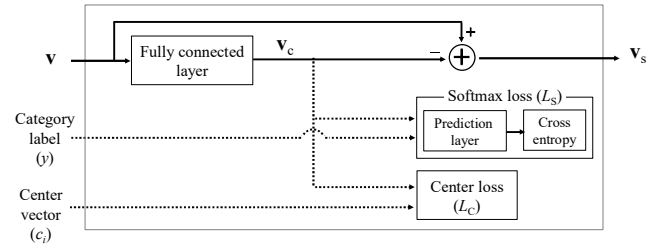


Fig. 2. The overall framework of SFE layer. Losses are drawn with dashed lines, input/output with solid lines. The proposed SFE layer first produces $\mathbf{v}_c$ using a fully connected layer, and then $\mathbf{v}_s$ predicting stylish matching clothes is obtained by removing $\mathbf{v}_c$ from $\mathbf{v}$.

layer. To train this layer, we use two different loss functions: the softmax loss function [13, 14], which forces $\mathbf{v}_s$ from different categories into staying apart, and the center loss function [15], which pulls each $\mathbf{v}_c$ to the center vector of its corresponding category. In other words, the softmax loss increases the inter-category difference and the center loss reduces the intra-category variation.

To calculate the softmax loss L_s , we first feed $\mathbf{v}_c$ into a prediction layer in order to obtain a probability distribution over category p , and then compute the cross entropy between p and category label y , which is defined as follows:

$$L_s = -y \log(p). \quad (3)$$

On the other hand, the center loss L_c penalizes the distance between $\mathbf{v}_c$ and its corresponding center vector, which is expressed as follows:

$$L_c = \frac{1}{2} \|\mathbf{v}_c^i - \mathbf{c}_i\|, \quad (4)$$

where $\mathbf{v}_c^i$ and $\mathbf{c}_i$ denote the category vector and center vector of the i -th category, respectively. A more detailed explanation about L_c can be found in [15].

Consequently, we modify the Siamese CNN by appending the SFE layer at the top of the base CNN as depicted in Fig. 3. To train the modified Siamese CNN, we intuitively define the loss function as follows:

$$L = L_{\text{contrastive}}(\mathbf{v}_{s,1}, \mathbf{v}_{s,2}) + \lambda_{\text{SFE}} \left(\frac{L_{s,1} + L_{s,2}}{2} + \frac{L_{c,1} + L_{c,2}}{2} \right), \quad (5)$$

where $L_{\text{contrastive}}(\cdot)$ is defined in equation (1), and λ_{SFE} is a hyper-parameter to control the balance between the $L_{\text{contrastive}}(\cdot)$ and the loss terms of SFE layer. The networks with the proposed loss function are trained to effectively predict stylish matching clothes.

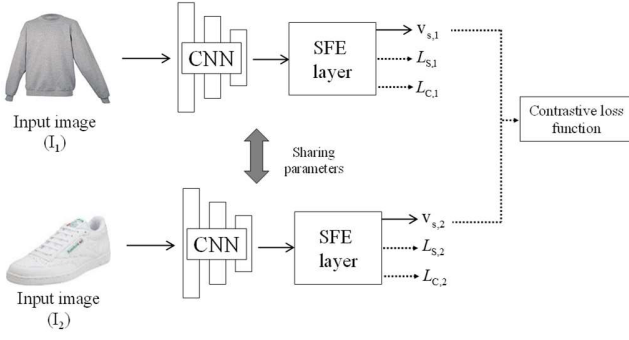


Fig. 3. The flow of modified Siamese CNN. Losses are drawn with dashed lines, input/output with solid lines. After training, the v_s is used of SMR system.

TABLE I
AUC COMPARISON OF THE CONVENTIONAL METHOD
WITH THE PROPOSED ONE

	Siamese CNN	Parameter-tuned Siamese CNN	Siamese CNN +SFE layer
GoogLeNet	0.8358	0.8469	0.8779
VGGNet	0.8457	0.8571	0.8762
MobileNet	0.8602	0.8629	0.8727

IV. PERFORMANCE ANALYSIS

A. Dataset and training

For our experiment, we employ the Amazon dataset [7] aggregating the preferences of Amazon customers for the relationship between pairs of clothing images. Since both positive and negative examples are required to train Siamese CNN, we generate the positive examples using pairs of clothing labeled as “also-bought” and “bought-together” in the Amazon dataset; on the other hand, we randomly sample the negative examples among pairs of clothing not having compatible labels. As described in [9], the same ratio of positive and negative examples in the training set are easy to classify. To avoid this problem, we generate a training set, having about 2 million examples, in which the ratio between the positives and negatives is 1:16. Also, we create a test set containing 117,586 pairs of positive and negative examples in a 1:1 ratio.

In order to train the Siamese CNN with SFE layer, we use GoogLeNet [13], VGG16 [14], and MobileNet [16], which are pre-trained on ILSVRC-2012-CLS [20], as a base network. The last 1000-dimensional fully connected layer of the base network is replaced with a 256-dimensional fully connected layer for generating the clothes vector. The SFE layer is appended to the replaced fully-connected layer. The constant value M in (1) is set to 100 in all our experiments. Since all the parameters in the proposed network are differentiable, we performed optimization using the Adam optimizer [21], which is a stochastic optimization method with the adaptive estimation of moments. Our network is trained for 200k iterations with a base learning rate of 10^{-4} and a batch size of 20 in an end-to-

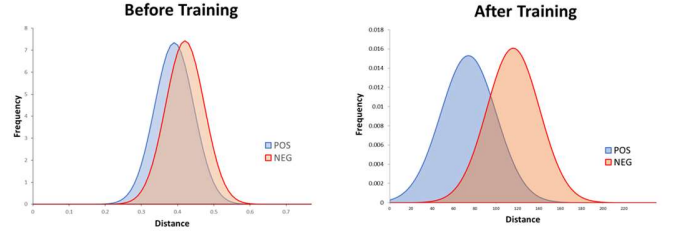


Fig. 4. Distributions of distances between two output vectors of our network for positive and negative examples for both before and after training.

TABLE II
INVESTIGATION OF THE HYPER-PARAMETER λ_{SFE}

Method	AUC
GoogLeNet [5]	0.8358
GoogLeNet [5] + SFE layer ($\lambda_{SFE}=1$)	0.8779
GoogLeNet [5] + SFE layer ($\lambda_{SFE}=10$)	0.8752
GoogLeNet [5] + SFE layer ($\lambda_{SFE}=100$)	0.8695

end manner. Our experiments were conducted on a single Titan X GPU (Pascal architecture) and implemented in *Tensorflow*.

B. Performance Evaluation

The objective of our network is to project the positive pairs close together and the negative pairs far apart in the feature space. In order to show the representational ability of our network, as shown in Fig. 4, we plot the distributions of the distance between the two style vectors of a pair of input examples before and after training [9]. This plot shows that the proposed network learned to separate positive and negative examples with a large margin. In addition, we measured the link prediction performance of the proposed network on the test set and compared it with the performance of the conventional Siamese CNN. Following the previous works [9, 12], we drew a receiver operating characteristic (ROC) curve, which is computed by sweeping a threshold value of vector distance, in order to predict if a pair of clothes was matching or non-matching and measured the performance using the area under the ROC curve (AUC). The larger the AUC value, the better the performance. Table 1 lists the performance in terms of AUC. These results show that the proposed network clearly outperforms the conventional Siamese CNN with GoogLeNet [13], VGG16 [14], and MobileNet [16] as its base network. In addition, to compare the proposed and conventional networks under the same total number of model parameters, we measured the performance of a parameter-tuned Siamese CNN which appends another 256-dimensional fully connected layer to the fully connected layer generating the clothes vector. As shown in Table 1, the parameter-tuned Siamese CNN shows better performance than the standard Siamese CNN in terms of AUC, but it does not perform better than the proposed network. This indicates that, when compared to the Siamese CNN having the same number of parameters, more accurate style information of



Fig. 5. Results of fashion recommendation. Each column shows the recommended outfits using the proposed method. Query images are indicated by a blue border. The other images are recommended fashion items.

clothing can be learned through the proposed SFE layer. Furthermore, we conduct experiments in order to confirm how λ_{SFE} influences performance, by adjusting the values of λ_{SFE} . In this experiment, we utilize GoogLeNet as a base network. As shown in Table 2, the SFE layer is not sensitive to λ_{SFE} for extracting style information. Therefore, we expect that the SFE layer can be easily used in a variety of SMR systems in order to obtain the clothing style.

C. Fashion recommendation

We recommend suitable outfits using the method in [9]. In particular, the outfits are generated as follows: Using the k -means clustering algorithm, we first cluster the style vectors of clothes in each class. We use $k = 20$ in our experiments and obtain 20 centroids $\{C_1, \dots, C_{20}\}$. Then, we find the nearest centroid C_n to the style vector of certain clothes. After that, the m closest items belonging to the n -th class are picked as candidates for matching clothes. We choose $m = 10$. Finally, we select an item among the candidates, which has the closest style vector to the style vector of certain clothes. Fig. 5 shows the outfits recommended by this algorithm.

V. CONCLUSION

In this paper, a novel feature layer used to extract the style information of clothing, called SFE layer, has been proposed. When compared with the conventional state-of-the-art network, the network using the proposed SFE layer shows superior performance in terms of link prediction. As a simple CNN layer, it is expected that the proposed SFE layer is compatible with all popular CNN architectures and is easy to implement. Therefore, we expected that our SFE layer can be applied to various SMR systems for suggesting more reliable clothing matches to customers.

REFERENCES

- [1] Y. Hu, X. Yi, and L. S. Davis, "Collaborative fashion recommendation: A functional tensor factorization approach," in Proc. ACM Int. Conf. Multimedia, 2015, pp. 129–138.
- [2] H. Lee, J. Seol, and S.-G. Lee, "Style2Vec: Representation Learning for Fashion Items from Style Sets," arXiv preprint arXiv:1708.04014.
- [3] E. Simo-Serra and H. Ishikawa, "Fashion style in 128 floats: Joint ranking and classification using weak data for feature extraction," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 298–307.
- [4] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2-16, pp. 1096–1104.
- [5] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan, "Deep domain adaptation for describing people based on fine-grained clothing attributes," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5315–5324.
- [6] W. Di, C. Wah, A. Bhardwaj, R. Priamuthu, and N. Sundaresan, "Style finder: Fine-grained clothing style detection and retrieval," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 8–13.
- [7] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2015, pp. 43–52.
- [8] R. He, C. Packer, and J. McAuley, "Learning compatibility across categories for heterogeneous item recommendation," in Proceedings of the IEEE conference on Data Mining, 2016, pp. 937–942.
- [9] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, "Learning visual clothing style with heterogeneous dyadic co-occurrences," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4642–4650.
- [10] Y.-S. Shih, K.-Y. Chang, H.-T. Lin, and M. Sun, "Compatibility family learning for item recommendation and generation," arXiv preprint arXiv:1712.01262.
- [11] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, "Visually-aware fashion recommendation and design with generative image models," arXiv preprint arXiv:1711.02231.
- [12] Q. Liu, S. Wu, and L. Wang, "DeepStyle: Learning User Preferences for Visual Recommendation," in Proc. ACM SIGIR Int. Conf. Research and development in information retrieval, 2017, pp. 841–844.
- [13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolution," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
- [14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556, 2014.
- [15] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for Deep face recognition," in Proc. Eur. Conf. Comput. Vis., 2016, pp. 499–515.
- [16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861.
- [17] R. Handseel, S. Chopar, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 1735–1742.

- [18] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in Proc. 14th Eur. Conf. Comput. Vis., Amsterdam, The Netherlands, Oct. 2016.
- [19] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015.
- [20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Rei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 1–42, Apr. 2015.
- [21] D. Kinga and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.