Outfit Recommendation with Deep Sequence Learning

Yangbangyan Jiang[1,2], Qianqian Xu[3], Xiaochun Cao[1,2,*]
[1]*State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*
[3]*Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS, Beijing, China*
{jiangyangbangyan,caoxiaochun}@iie.ac.cn, xuqianqian@ict.ac.cn

***Abstract*—Choosing a proper clothing collocation requires the sense of fashion. Yet modeling how people select items is challenging: the items in a collocation should be compatible but there are too many attributes to consider (e.g., color, texture, style) for each kind of fashion items. In this paper, we propose to learn a global compatible outfit generation model from existing outfit images and text descriptions. Our approach relies on a bidirectional LSTM to model the relationship between different categories of fashion items and then predict the item based on all the other items. Meanwhile, embedded visual semantic descriptions are exploited to guide the generation with attribute information. Combining these structures, it is guaranteed that in the resulting outfit, items share a similar style and neither redundant nor missing items exist for essential categories. We demonstrate our method applied to an outfit dataset containing about 160,000 fashion items. Experimental results indicate that a good sense of fashion is obtained by the proposed method.**

***Index Terms*—Outfit recommendation, Deep learning, Bi-LSTM**

I. Introduction

With the rise of the consuming capacity and will of consumers, people are spending more time and money on dressing up themselves. On account of the huge potential business opportunities lying in the fashion domain, this field has gained much attention in recent years. Various applications appear to solve subproblems in this area. Some studies focus on clothing parsing [1]. [2] utilizes a retrieval-based approach to deal with parsing. [3] proposes a clothing co-parsing framework to segment and label fashion images. There are also some work on matching street and shop item images using parts alignment [4], [5] and deep learning techniques [6]. And some studies aim at learning fashion representation for clothing attributes and landmarks [7], [8].

Besides the applications above, researchers attempt to address the problem of fashion item and outfit recommendation in several ways. In this problem, it is important to measure the compatibility of fashion items to choose the best candidate for query items. Such score can be calculated using Siamese networks [9] or user behavior co-occurrence [10]. On the other hand, the popularity of items learned from online outfit data could also be such a metric [11], [12]. Based on these studies, some outfit generation algorithms are proposed. To name a few, [13] implements a latent SVM framework with manually labeled attributes to generate fashion items based on occasions. [14] proposes a collaborative approach to recommend outfit items, which model the connections between users and fashion items by functional tensor factorization.

Most existing researches on scoring the compatibility of candidate items focus on pairwise compatibility. However, such methods often tend to emphasize a specific part of the items while ignoring the global dependency. Moreover, the high computational cost of enumerating all the pairs for a large candidate item set makes this kind of method less practical. Therefore, [12] exploits an RNN to predict the popularity of fashion outfits from the image and text features. Following this work, [15] directly models the global compatibility relationships among fashion items together, which improves the quality of generation. These studies preserve that fashion items in the same outfit should a share similar style, and neither redundant nor missing categories exist in the outfit(e.g., two pair of shoes, or no tops but only bottoms in the outfit).

Inspired by these methods, in this paper, we propose a novel end-to-end outfit recommendation algorithm, which aims at modeling item global compatibility and generating a fashion outfit for item images and text queries. We fix the order of items in an outfit in terms of their categories such that an outfit can be treated as a sequence and each item is a time step. Then we employ a bidirectional LSTM to mine the dependencies among items and perform sequence prediction. Specifically, the algorithm generates outfit items that maximize the co-occurrence probability of given items and themselves. Furthermore, we perform visual-semantic embedding for the images and text queries to provide attribute requirements for the outfit. Hence items in the generated result outfit share a similar style via the proposed method. Experimental results for various queries show the effectiveness of the proposed method.

II. Methodology

Most of existing outfit recommendation methods ignore the order of items in an outfit, while [15] takes advantage of

The research of Yangbangyan Jiang and Xiaochun Cao was supported by National Key Research and Development Plan (No.2016YFB0800603), National Natural Science Foundation of China (No.U1636214, 61650202, 61733007), Key Program of the Chinese Academy of Sciences (No.QYZDB-SSW-JSC003). The research of Qianqian Xu was supported in part by National Natural Science Foundation of China (No.61672514), Beijing Natural Science Foundation (4182079), Youth Innovation Promotion Association CAS, and CCF-Tencent Open Research Fund.

*Corresponding Author.

978-1-5386-5321-0/18/$31.00 ©2018 IEEE

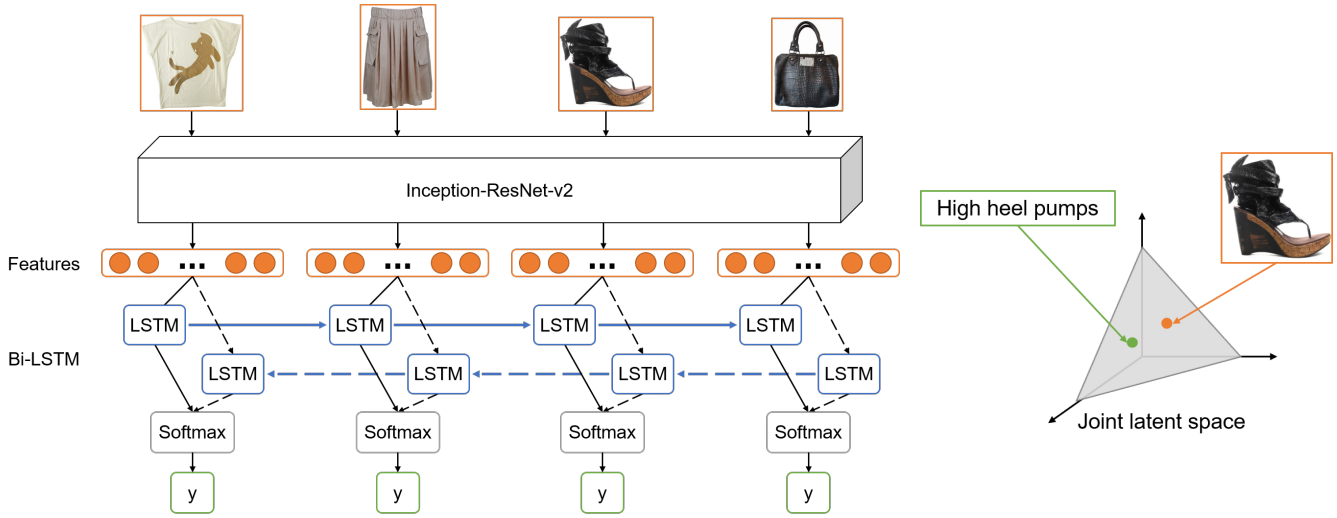

Fig. 1. Network architecture. (a) We use an Inception-ResNet-v2 convolutional network to obtain embedded feature vectors of images and then input the features into a bidirectional LSTM to generate the outfit item sequence. The forward LSTM learns from the past information and the backward LSTM learns from the future one. (b) We transform the image and text into a joint space to measure the similarity.

that to prevent the framework from enumerating all possible combinations of items. Accordingly, we also sort items in the outfit by their categories in the order of *top, bottom, bag*, and *accessories*, to cast each outfit into an ordered sequence. Then the generation could be turned into a sequence prediction problem, in which we should model the connection between different categories of items in the outfit. Long-short time memory network (LSTM) [16] is an effective solution for such problem. With its short-term memory of past information that lasts for a long period of time, LSTM can utilize all the past information to predict the next item. Furthermore, since we need to predict the items either before or after the query item in the sequence, we introduce a bidirectional LSTM (Bi-LSTM) [17] to address this bidirectional prediction problem. Such networks could learn from both past and future information through its forward and backward LSTMs.

Given a dataset with n outfits $\mathcal{D} = \{\boldsymbol{S}_i\}_{i=1}^{n}$, where $\boldsymbol{S}_i = \{(\boldsymbol{X}_k, \boldsymbol{R}_k)\}_{k=1}^{m}$ denotes the item set for i-th outfit, $\boldsymbol{X}_k$ and $\boldsymbol{R}_k$ denotes the k-th item image and text description in the outfit, we implement a deep neural network, consisting of a convolutional neural network (CNN), a Bi-LSTM and a visual-semantic embedding, to perform outfit generation. We first extract the latent feature representation of item images by the CNN, and then feed those features into the Bi-LSTM. In the Bi-LSTM, the forward LSTM learns from the past information and the backward one learns from the future information. Therefore, the Bi-LSTM can capture the compatibility of the entire outfit, i.e., the style shared by all the fashion items in the outfit. Meanwhile, to enable the network to generate items according to specific attributes, a visual-semantic embedding layer is exploited to introduce text information into the learning. The network architecture is illustrated in Fig. 1a.

In the following, we describe the function of these three modules.

CNN: This component is used for feature extraction and has no contribution to the objective. CNN's success in image processing and computer vision [18], [19] has demonstrated its promising ability to capture intrinsic features of images on large-scale datasets. Therefore, we exploit an Inception-ResNet-v2 model [20] to obtain the feature vector $\boldsymbol{v}_k$ for the k-th image $\boldsymbol{X}_k$. This model benefits from both residual connections [19] and Inception blocks [18], thus achieves better performance.

Bi-LSTM: This module learns the relationship among all the items. Through the CNN, each outfit item set $\boldsymbol{S}_i$ is transformed to $\boldsymbol{T}_i = \{\boldsymbol{v}_k\}_{k=1}^{m}$ then fed into Bi-LSTM as a sequence. Here, we add two zero vectors $\boldsymbol{v}_0$ and $\boldsymbol{v}_{m+1}$ to the set as the end marks for Bi-LSTM. Consequently, the sequence becomes $\tilde{\boldsymbol{T}}_i = \{\boldsymbol{v}_k\}_{k=0}^{m+1}$. At the t-th time step, the forward LSTM utilizes all the past information for prediction. Specifically, it outputs the probability of seeing $\boldsymbol{v}_t$ conditioned on all the previous items $\{\boldsymbol{v}_k\}_{k=1}^{t-1}$ by applying a softmax function to the former hidden state vectors $\boldsymbol{h}_{\mathrm{f},t-1}$ as follows:

$$Pr_{\mathrm{f}}(\boldsymbol{v}_t|\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{t-1}) = \frac{\exp(\boldsymbol{h}_{\mathrm{f},t-1}\boldsymbol{v}_t)}{\sum_{\boldsymbol{v}\in\mathcal{V}_C}\exp(\boldsymbol{h}_{\mathrm{f},t-1}\boldsymbol{v})} \tag{1}$$

where $\mathcal{V}_C$ denotes all the items in the current batch.

We choose the item with the highest occurrence probability given all previous items as the generation result. Then we can formulate the loss function for the forward LSTM as follows:

$$\ell_{\mathrm{f}} = -\frac{1}{m}\sum_{t=2}^{m+1}\log Pr_{\mathrm{f}}(\boldsymbol{v}_t|\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{t-1}) \tag{2}$$

Likewise, the backward LSTM calculates the probability of seeing $\boldsymbol{v}_t$ conditioned on all the future items $\{\boldsymbol{v}_k\}_{k=t+1}^{m+1}$ using the hidden state vectors $\boldsymbol{h}_{\mathrm{b},t+1}$ as follows:

$$Pr_{\mathrm{b}}(\boldsymbol{v}_t|\boldsymbol{v}_{t+1}, \cdots, \boldsymbol{v}_m) = \frac{\exp(\boldsymbol{h}_{\mathrm{b},t+1}\boldsymbol{v}_t)}{\sum_{\boldsymbol{v}\in\mathcal{V}_C}\exp(\boldsymbol{h}_{\mathrm{b},t+1}\boldsymbol{v})} \tag{3}$$

And the loss function for the backward LSTM is formulated as follows:

$$\ell_{\mathrm{b}} = -\frac{1}{m}\sum_{t=m-1}^{0} \log Pr_{\mathrm{b}}(\boldsymbol{v}_t|\boldsymbol{v}_{t+1},\cdots,\boldsymbol{v}_m) \tag{4}$$

Hence the compatibility loss of each outfit can be is the sum of the two loss above:

$$\ell_i = \ell_{\mathrm{f}} + \ell_{\mathrm{b}} \tag{5}$$

Visual-semantic Embedding: This module integrates text information into the model to provide certain attribute queries for the recommendation. Item images and corresponding text descriptions are mapped onto a joint latent space, i.e., the visual-semantic space, such that the similarity between an image and a text can be estimated, as depicted in Fig. 1b.

Given a pair of image $\boldsymbol{X}$ and text $\boldsymbol{R} = \{\boldsymbol{r}_j\}_{j=1}^{l}$, where $\boldsymbol{r}_j$ denotes the j-th word in the text, we first transform the image and each word into feature vectors $\boldsymbol{v}$ and $\boldsymbol{c}_j$ using the CNN and TF-IDF algorithm, respectively. Then the text is represented by $\boldsymbol{c} = \frac{1}{l}\sum_{j=1}^{l}\boldsymbol{c}_j$. We learn an image projection matrix $\boldsymbol{W}_{\mathrm{I}}$ to project the image feature $\boldsymbol{v}$ into the joint space and obtain the image latent representation by $\boldsymbol{i} = \boldsymbol{W}_{\mathrm{I}}\boldsymbol{v}$. Likewise, we transform the text representation $\boldsymbol{c}$ into the joint space by $\boldsymbol{t} = \boldsymbol{W}_{\mathrm{T}}\boldsymbol{c}$, where $\boldsymbol{W}_{\mathrm{T}}$ is the text projection matrix. Then we use the normalized cosine distance to measure the similarity between the image and text:

$$s(\boldsymbol{i},\boldsymbol{t}) = \frac{\boldsymbol{i}\cdot\boldsymbol{t}}{|\boldsymbol{i}|\cdot|\boldsymbol{t}|} \tag{6}$$

where $|\cdot|$ denotes the length of the vector.

Then we calculate the loss of the embedding for the entire dataset using the contrastive loss as follows:

$$\begin{aligned}\ell_{\mathrm{emb}} = &\sum_{\boldsymbol{i}\in\mathcal{I}}\sum_{\boldsymbol{t}_{\mathrm{non}}\in\mathcal{T}}\max\{0,\zeta - s(\boldsymbol{i},\boldsymbol{t}) + s(\boldsymbol{i},\boldsymbol{t}_{\mathrm{non}})\}+\\ &\sum_{\boldsymbol{t}\in\mathcal{T}}\sum_{\boldsymbol{i}_{\mathrm{non}}\in\mathcal{I}}\max\{0,\zeta - s(\boldsymbol{i},\boldsymbol{t}) + s(\boldsymbol{i}_{\mathrm{non}},\boldsymbol{t})\}\end{aligned} \tag{7}$$

where $\mathcal{I}$ and $\mathcal{T}$ are the representation set for all the images and text respectively, $\boldsymbol{i}_{\mathrm{non}}$ denotes the image representation that does not match the text $\boldsymbol{t}$, $\boldsymbol{t}_{\mathrm{non}}$ denotes the text representation that does not match the image $\boldsymbol{i}$, and ζ is the margin.

Putting all these together, we finally get the overall objective of our network for the entire dataset:

$$\mathcal{L} = \sum_{i=1}^{n}\ell_i + \ell_{\mathrm{emb}} \tag{8}$$

In training, each outfit is fed into the network to learn the dependency among different categories of items. For recommendation, some query images and text descriptions are given and we aim at generating missing items to build a proper outfit. To this end, firstly, we input the first query item and generate an initial item set via the Bi-LSTM. When there are more than one query items, we find the nearest neighbor of the second item in the initial set and replace that with the query one. If there is a gap between the first and second item, we select the item that maximizes the co-occurrence probabilities of current items given items before and after the gap position to fill it. After that, we run the inference in both directions to generate a new outfit. We repeat this process until every query item exists in the outfit. Note that the step of filling blank positions between two query items guarantees the compatibility of the outfit subsequence, consequently the global compatibility of the result.

III. Experiment

A. Dataset

We use the dataset collected from Polyvore [1], a fashion website providing the outfits created by users [15]. The Polyvore dataset contains 21,889 outfits and 164,379 items. Each outfit consists of up to 8 fashion items with distinct categories such as *Coats, Pants, Hats*. These outfits are split into 17,316 for training, 1,497 for validation and 3,076 for testing. The order of items in each outfit is fixed to *tops, bottoms, shoes*, and *accessories*. Specifically, for accessories, the order is usually like handbags, hats, glasses, watches, necklaces, earrings, etc. This property allows the Bi-LSTM to learn the relationship between the items in an outfit.

B. Settings

We extract a 1792-D feature vector for each image by the Inception-ResNet-v2 model and transform it into a 512-D vector as the input of Bi-LSTM using a fully-connected layer. Note that the feature obtained by the CNN is dropouted thus more exquisite and useful. In the Bi-LSTM, the number of hidden units is set to 512 and the dropout rate is 0.7. Meanwhile, the dimension of joint space representation is also 512 and the margin ζ is set to 0.2. For training, the batch size is set to 2, namely, there are 2 outfits in each mini-batch. Since we use a pre-trained Inception-ResNet-v2 model[2], the learning rate is set to 0.0005 and decayed by a factor of 2 every 2 epochs. The network is implemented using TensorFlow [21] and run on a server with an NVIDIA Titan X GPU.

C. Results

Some generation results are demonstrated in Fig. 2. In these examples, we show the results for queries combining items like *tops, bottoms, hats, shoes*, and attributes like *color, occasion, gender*. For instance, in Fig. 2a, we select a jumper and a pair of sneakers with a "gray" attribute requirement. Accordingly, the result includes a hollow necklace, a sports backpack, and a baseball cap, all of which share a mainly gray tone. When there is a single query item, e.g., a baseball cap in Fig. 2c, our model could generate corresponding browser, trousers, sneakers, backpack, and even bracelet and rings with a hip pop style. Fig. 2b and 2d also show compatible examples, in which all the items are very leisure or free. On the other hand, some results are not very pleasing. If we change the trousers to a baseball cap in the query items and set the query text to

[1]https://polyvore.com

[2]https://github.com/tensorflow/models/tree/master/research/slim

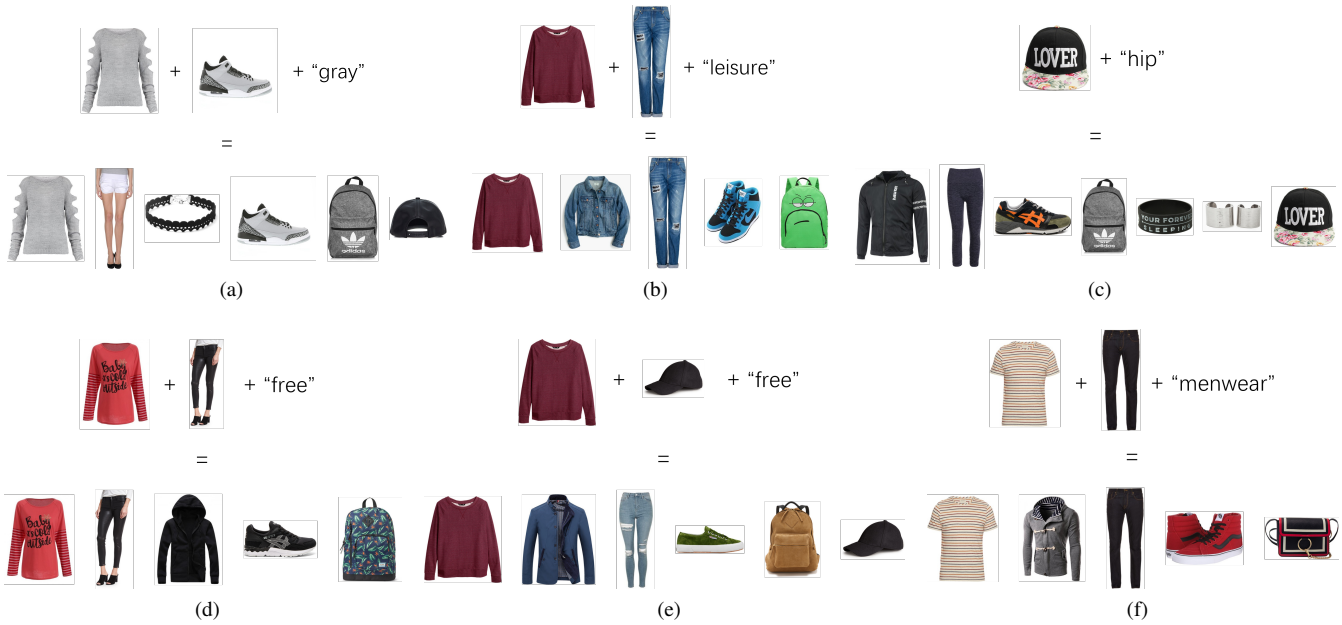

Fig. 2. Recommendation results for the network. (a)-(d) show good examples and (e)-(f) are bad ones.

“free”, then we get the generated outfit illustrated in Fig. 2e. It is easy to see that the trousers are designed for women while the coat is a menswear, thus the target gender of this outfit is a little confusing. Fig. 2f demonstrates another gender mistake. In this outfit, the messenger bag is clearly suitable for women, which is not keeping with our query text “menwear”. Meanwhile, the style of the shoes and bag is not compatible with that of the others, since the former is very modern but the latter is classical.

IV. Conclusion

In this paper, we propose a novel outfit recommendation method. Through sorting the items in each outfit in terms of their categories, we treat an outfit as a sequence and then cast outfit recommendation into a sequence prediction problem. Seeing each fashion item in the outfit as a time step, we implement a bidirectional LSTM to capture the style shared by the items and predict the next item based on all the previous ones. Meanwhile, a visual-semantic embedding layer is exploited to specify certain attribute requirement for the outfit, in which the image and text representations are both projected to a joint latent space thus the similarity can be measured. Experiment results on a large dataset demonstrate that our proposed method can generate natural and proper outfits according to various queries.

References

[1] X. Liang, L. Lin, W. Yang, P. Luo, J. Huang, and S. Yan, “Clothes co-parsing via joint image segmentation and labeling with application to clothing retrieval,” *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1175–1186, 2016.

[2] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, “Retrieving similar styles to parse clothing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 5, pp. 1028–1040, 2015.

[3] W. Yang, P. Luo, and L. Lin, “Clothing co-parsing by joint image segmentation and labeling,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3182–3189.

[4] Z. Cheng, X. Wu, Y. Liu, and X. Hua, “Video2shop: Exact matching clothes in videos to online shopping images,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4169–4177.

[5] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3330–3337.

[6] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, “Where to buy it: Matching street clothing photos in online shops,” in *IEEE International Conference on Computer Vision*, 2015, pp. 3343–3351.

[7] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “Deepfashion: Powering robust clothes recognition and retrieval with rich annotations,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1096–1104.

[8] Q. Chen, J. Huang, R. S. Feris, L. M. Brown, J. Dong, and S. Yan, “Deep domain adaptation for describing people based on fine-grained clothing attributes,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5315–5324.

[9] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, “Learning visual clothing style with heterogeneous dyadic co-occurrences,” in *IEEE International Conference on Computer Vision*, 2015, pp. 4642–4650.

[10] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. J. Belongie, “Learning visual clothing style with heterogeneous dyadic co-occurrences,” in *IEEE International Conference on Computer Vision*, 2015, pp. 4642–4650.

[11] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun, “Neuroaesthetics in fashion: Modeling the perception of fashionability,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 869–877.

[12] Y. Li, L. Cao, J. Zhu, and J. Luo, “Mining fashion outfit composition using an end-to-end deep learning approach on set data,” *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1946–1955, 2017.

[13] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, “Hi, magic closet, tell me what to wear!” in *ACM Multimedia Conference*, 2012, pp. 619–628.

[14] Y. Hu, X. Yi, and L. S. Davis, “Collaborative fashion recommendation: A functional tensor factorization approach,” in *ACM Multimedia Conference*, 2015, pp. 129–138.

[15] X. Han, Z. Wu, Y. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *ACM Multimedia Conference*, 2017, pp. 1078–1086.
[16] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.
[17] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 273–278.
[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI Conference on Artificial Intelligence*, 2017, pp. 4278–4284.
[21] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.