A Visual Similarity Recommendation System using Generative Adversarial Networks

Betul AY
Computer Engineering Department
Firat University
Elazig, Turkey
betulay@firat.edu.tr

Zeynep KOYUN
Big Data and Artificial Intelligence Laboratory
Firat University
Elazig, Turkey
zeynep@bigailab.firat.edu.tr

Mehmet DEMIR
IT R&D Department
FLO Magazacilik A.S.
Istanbul, Turkey
mehmet.demir@flo.com.tr

Galip AYDIN
Computer Engineering Department
Firat University
Elazig, Turkey
gaydin@firat.edu.tr

***Abstract*— The goal of content-based recommendation system is to retrieve and rank the list of items that are closest to the query item. Today, almost every e-commerce platform has a recommendation system strategy for products that customers can decide to buy. In this paper we describe our work on creating a Generative Adversarial Network based image retrieval system for e-commerce platforms to retrieve best similar images for a given product image specifically for shoes. We compare state-of-the-art solutions and provide results for the proposed deep learning network on a standard data set.**

Keywords—image retrieval, deep learning, image similarity

I. Introduction

The Content-based Image Retrieval (CBIR) [1] is a process that takes a query image and find relevant images from a large database of target images. CBIR systems help users retrieve similar images based on their visual content features such as color, shape, volume, texture, local geometry and other information. There are numerous application areas that use these systems from past to present such as art galleries management, architectural, engineering and interior design, geographic information systems, weather forecasting, retail systems, fashion design, trademark database management, medical image management and other e-commerce applications [2].

There are two basic steps for CBIR process: Feature vector extraction of each image and similarity calculation between the image vectors. The success of these systems will vary depending on the accuracy and reliability of the feature vector representing the image. Consequently, it is one of the most challenging tasks to provide a fast, accurate and efficient model to extract an automatic feature vector of target images. Another challenge is to label a large amount of training data. Supervised training for all target images limits the generalizability of the learned deep representations to new classes [3]. To overcome these limitations, the interest towards semi-supervised and unsupervised learning techniques has increased.

Generative Adversarial Networks, known as GAN, was introduced by Ian Goodfellow [4] to address the problem of unsupervised learning in 2014. Since GANs learn deep representations using unlabeled training data, they are currently one of the most popular and emerging techniques for semi-supervised and unsupervised learning. GANs are composed of two deep neural networks called generator and discriminator. The generator network takes random noise as input and generates a realistic image as output. The discriminator network is a regular neural network classifier which tries to calculate the probability that the input is real or fake. GAN has come up with very good and promising results recently to generate visually realistic images. But GANs, the creative power of artificial intelligence, have not limited to only generating realistic images. Some applications of this networks are image-to-image translation (CycleGAN [5]), high quality image generation from low quality images (SRGAN [6]), image generation from text (StackGAN [7]), discovering cross-domain relations transformation such as fashion items (DiscoGAN [8]), facial makeup transfer (Beautygan [9]) and other applications reviewed in [10]. Hou et al. [11] have used GANs with a pretrained convolutional neural network (VGGNet [12]) to extract deep features from generated images. Their system consists of three networks: Generator, VGGNet and discriminator. In their system, generator network doesn't directly feed the real and fake images to the discriminator. Convolutional features extracted from the pretrained model are fed to the discriminator network. When they compared the generated images considering clear facial parts with DCGAN [13] and DFC-VAE [14], their model has generated more realistic face images.

Visual search and recommendations are crucial for e-commerce sites. Traditional recommender systems based on collaborative filtering often face problems of computational difficulty, scalability, and sparsity on a large amount of data. Moreover, these systems use click and purchase history of the user to recommend a new product by ignoring the image content. In this paper, we present a content-based visual recommendation system that can be used by the potential client and best recommendation for the target client. Our main contributions are development and evaluation of deep learning based image retrieval techniques by providing comparative study for shoe fashion items recommendation.

The rest of the paper is structured as follows. We briefly reviewed related work in Section II and present background on Generative Adversarial Networks used for this paper in Section III. In Section IV, the proposed network architecture and the train experiments are described in detail. We discuss and analyze the experimental results in Section V and draw conclusion in Section VI.

II. Related Work

Due to the great interest in e-commerce applications, recently image retrieval techniques for fashion items have gained huge popularity. Many studies have used machine learning and predictive analysis to retrieve and recommend fashion items for each customer [15-19]. However, research highlights some of the challenges in retrieving images for the users due to the fact that the fashion concept is subtle and subjective for the human vision evaluation. Therefore, content-based Image Retrieval for e-commerce platforms is still an open problem to reach a general consensus [20].

978-1-7281-2914-3/19/$31.00 ©2019 IEEE
DOI 10.1109/Deep-ML.2019.00017

Kiapour et al. [21] have developed deep learning baseline methods for exact street to shop retrieval. The goal of the study is to find similar clothing items in an online shop for a given real-world photo which contains clothing items. Another study [22] addresses the problem of cross-domain fashion product retrieval by trying to retrieve similar clothing items from online shopping images. Khosla and Venkataraman [23] has used convolutional neural networks (CNN) to address the retrieval problems on a dataset including over 30,000 shoe images. They have achieved 75.6% precision with pre-trained VGGNet model on a shoe dataset that is re-scraped from zappos.com (instead of using the UT-Zap50K dataset). For each image in the dataset, they have utilized feature vectors extracted from the last fully connected layer of pre-trained VGGNet model. Shankar et al. [24] have presented a visual search and recommendation system for e-commerce. Their system uses a deep CNN to learn image embeddings of fashion products. Since the network needs to label data for the training dataset, they have created a large annotated dataset by labelling images collected from the Fashionista dataset [25] and Flipkart catalog images. In this paper, we present a similar recommender system that retrieve a ranked list of shoe images similar to queried shoe image with different deep neural networks. For this task, we trained the proposed network from scratch with 67,000 shoe images collected from two major Turkish e-commerce sites: flo.com.tr and trendyol.com.tr. To compare the proposed neural network with existing pre-trained models in terms of time and performance, we use UT-Zap50K [26] as standard benchmark dataset. The proposed system is easy to extend due to the fact that we do not require labeled images in the training dataset, in other words the major advantage of the proposed system is that we can easily extend and improve the model by simply adding new product images into the training images folder.

III. Background On Generative Adversarial Networks

A. GAN

Vanilla GANs, shown in Fig.1, are made up of two distinct networks: Generator G and discriminator D. While the goal of G is to map random noise $z \in \mathbb{R}^Z$ to generate inputs $G(z)$, the target of D is to predict a probability of an image being real $D(x)$ or fake $D(G(z))$ by taking a dataset including real images x={x^1,…,x^N } and fake images $G(z)$ generated from G.

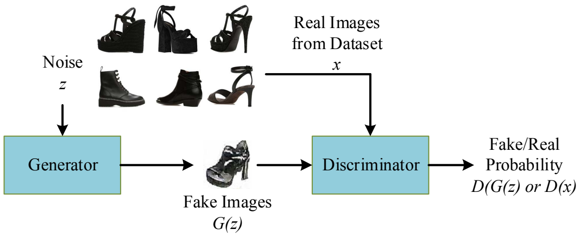

Fig. 1. Vanilla GAN architecture

More formally, the minimax objective of GAN $\underset{G}{min}\ \underset{D}{\max} V(D,G)$ is given in the following expression [4]:

$$V(D,G) = \mathbb{E}_{x\sim P_{data}}[logD(x)] + \mathbb{E}_{z\sim P_{noise}}\left[\log\left(1-D\left(G(z)\right)\right)\right] \quad (1)$$

Where P_{data} is real data distribution and P_{noise} is the noise distribution that can be named as model distribution.

B. DCGAN

Deep Convolutional GAN, typically called DCGAN, have also two networks named as generator and discriminator like GANs. The generator network tries to fool the discriminator network with fake images and the discriminator network tries to correctly classify images as real or fake. But for more complex tasks, this architecture uses deep convolutional networks composed of transposed convolutional layers for the generator and discriminator unlike vanilla GANs. Fig. 2 illustrates the architecture used in the original DCGAN paper:

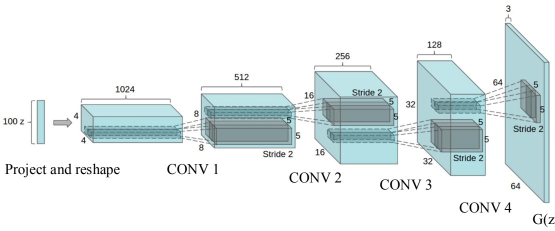

Fig. 2. DCGAN generator architecture [13]

In DCGAN, the input of the generator will be a noise vector z and the output of the network will be a $tanh$ output. The transposed convolutional layers generate new images. There are no fully connected or pooling layers in the architecture. The architecture has replaced the pooling layers such as max-pooling with strided convolutions (discriminator) and fractional-strided convolutions (generator).

C. InfoGAN

Information Maximizing Generative Adversarial Networks (InfoGAN), an extension of GANs, learn a disentangled representation by decomposing the input noise vector into two parts: a source of incompressible noise z and a latent code c. The method proposed in [26] discovers latent factors of variation by maximizing the mutual information between c and generator distribution $G(z,c)$. By adding regularization term with a hyper-parameter λ, information-regularized minimax objective of InfoGAN $\underset{G}{min}\ \underset{D}{\max} V_I(D,G)$ is formulated as follows:

$$V_I(D,G) = V(D,G) - \lambda \mathrm{I}(\mathrm{c}; \mathrm{G}(\mathrm{z},\mathrm{c}) \quad (2)$$

Where I(c; G(z, c)) is the mutual information term. Since the calculation of this term is computationally complex, they use lower bounding mutual information technique. The following final objective of InfoGAN $\underset{G,Q}{min}\ \underset{D}{\max} V_{InfoGAN}(D,G,Q)$ defines this solution with a variational lower bound $L_I(G,Q)$:

$$V_{InfoGAN}(D,G,Q) = V(D,G) - \lambda L_I(G,Q) \quad (3)$$

IV. Proposed Network Architecture

In this study, we aim to build a visual similarity recommendation system. Our work consisted of two major steps: Firstly, we explore the best methods to learn deep feature representations extracted from a given image. For this purpose, we follow the general idea of InfoGAN [27], which has proven successful for generating realistic images and meaningful representations based on mutual information. We also use the learned deep representations of popular pre-trained models by removing the last layer for the shoe recommendation task. Secondly, we make a simple distance calculation between the extracted features of the query image and other images in dataset. Finally, we compare the proposed network which is trained from scratch with several pre-trained model architectures including Densenet [28], Resnet [29], Inception-V3 [30], MobileNet [31] and VGGnet [32]. The proposed network architecture is inspired by InfoGAN [27] and includes several changes to the original InfoGAN model which are shown in Table 1.

TABLE I. THE DISCRIMINATOR AND GENERATOR NETWORKS USED FOR SHOE DATASET

Discriminator Model D / Recognition Network Q	**Generator Model G**
Input 128x128 Color image	Input ∈ □^{108}
3x3 conv2d. 16 IRELU. stride 2.	FC. 4x6x256
Dropout (.5)	3x3 conv2d_transpose. 128 RELU. stride 1. batchnorm
3x3 conv2d. 32 IRELU. stride 1. batchnorm	Dropout (0.6)
Dropout (.5)	3x3 conv2d_transpose. 64 RELU. stride 1. batchnorm
3x3 conv2d. 64 IRELU. stride 2. batchnorm	Dropout (0.6)
Dropout (.5)	3x3 conv2d_transpose. 32 RELU. stride 1. batchnorm
3x3 conv2d. 128 IRELU. stride 1. batchnorm	Dropout(0.6)
Dropout(.5)	3x3 conv2d_transpose. 16 RELU. stride 1. batchnorm
3x3 conv2d. 256 IRELU. stride 2. batchnorm	Dropout(0.6)
Dropout(.5)	3x3 conv2d_transpose. 16 RELU. stride 1. batchnorm
3x3 conv2d. 16 IRELU. stride 1. batchnorm	Dropout(0.6)
FC. 1 sigmoid for D	3x3 conv2d_transpose. 3 Tanh. stride 1.
FC. 8 Tanh for Q	Dropout(0.6)

The input of discriminator model D is a color image (128-pixel x 128-pixel x 1-channel). The input of generator model G is a concatenated vector with dimension 108, that consists of noise variable (100) and latent code (8) which represents the class information. We don't actually have a labeled data (categorical code is zero), but we assume that our data basically consist of 8 different classes (heels, sandals, sports, boots, high boots, loafers, slippers and flats), so the latent code value is set to 8. For the discriminator: (1) We apply Batch Normalization [33] except for the first layer to stabilize learning. (2) Dropout [34] is applied to all layers with 50% dropout rate to prevent overfitting and memorization problems. (3) Leaky ReLU is used after all convolution layers. (4) At the last layer, sigmoid activation function is used to get the probability of whether an image is real or fake. For the generator we use following techniques: (1) Batch Normalization is applied to all layers. (2) We apply dropout after all layers with 60% dropout rate because of the fact that more realistic images are generated with low dropout values (dropout < 1). (3) ReLU after each transposed convolutional layers and Tanh function after the output layer are used as explained in [13]. D and Q share the same network structure except for the output units on the last layer [27]. At the last layer, Q uses tanh activation function.

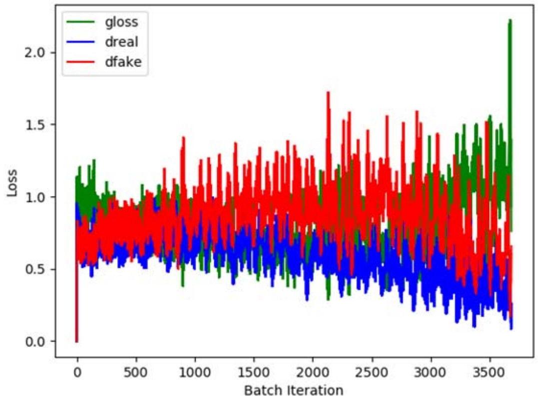

Fig. 3. Loss results for proposed model

The training results of the proposed network is presented in Fig. 3. The points where the gloss (generator loss) value is lower than the dreal (discriminator real loss) value at approximately 500-2500 intervals are the ideal points for us. Because at these points the gloss values are smaller than the dreal values. In other words, generator has started to generate images similar to the real images in the dataset. For this reason, the discriminator is unable to distinguish between the fake images and the real images. After approximately 2500 iteration, it is observed that the difference between the dfake (discriminator fake loss) values and the dreal values decreases. This situation also tells us that the generator began to produce data similar to the real data. Hence the fake images started to resemble the real images. Therefore, the checkpoints recorded at these points should also be tested.

V. EXPERIMENTS AND RESULTS

TABLE II. PERFORMANCE COMPARISON FOR THE MODELS USED IN THIS STUDY

Model	**Size (MB)**	**Inference Time (sec)**	**Precision (%)**
VGG16	528 MB	0.22	0,75
VGG19	549 MB	0.21	0,81
ResNet50	98 MB	0.09	0,79
ResNet101	171 MB	0.16	0,80
ResNet152	232 MB	0.23	0,82
InceptionV3	92 MB	0.11	0,64
InceptionResNetV2	215 MB	0.06	0,51
MobileNet	16 MB	0.03	0,44
MobileNetV2	14 MB	0.04	0,73
DenseNet121	33 MB	0.10	0,61
DenseNet169	57 MB	0.12	0,64
DenseNet201	80 MB	0.15	0,56
The Proposed Model	23.7 MB	0.004	0,84

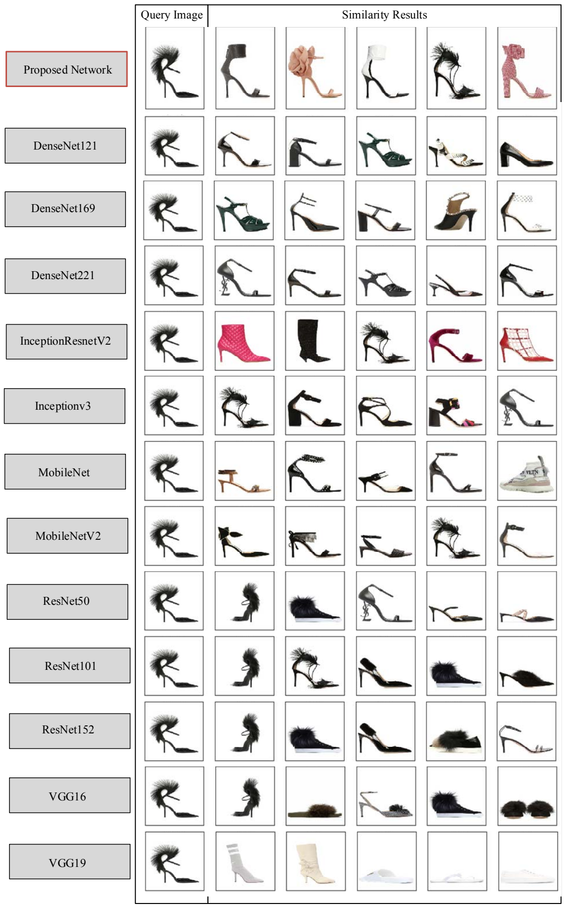

Fig. 4. Comparative results for the state-of-the-art pre-trained models with the proposed model

To train the proposed deep neural network and conduct the performance tests we use the same server which has 24-core Intel Xeon E5-2628L CPU, 256 GB RAM and runs Ubuntu Server 16.04 OS. The server also has 8 NVidia GTX 1080-Ti GPUs which are only used in the training steps. Tensorflow is used as the framework for running the models. The models are tested on CPU with a mini-batch size of 1.

The output of the discriminator for the proposed model gives us a feature vector with a size of [1, 1536]. We store these vector values in the form of arrays on MongoDB database. We use the Euclidean distance to calculate the similarity scores between query image and retrieved images.

Table 2 presents the test results including model size, inference time for feature vectors and precision rates for the proposed generative network and other popular convolutional neural network based pre-trained models. We conducted the tests on 10,000 randomly selected shoe images from UT-Zap50K benchmark dataset.

The major problem we tackle in this study is the image retrieval problem, which can be summarized as; given a query image, retrieving similar or relevant images from a dataset. The success of any retrieval solution is hard to calculate due to the fact that the concept of image similarity is highly subjective. Therefore, to calculate performance of the CBIR systems several approaches have been proposed in the literature. In this study we use the Standard Precision Metric which is formulated as:

$$Precision = \frac{\#\ of\ relevant\ items\ retrieved}{\#\ of\ retrieved\ items} \qquad (4)$$

To calculate the precision values for the models we use 8 different classes exist in the Zap50K dataset: heels, sandals, sports, boots, high boots, loafers, slippers and flats. To count the relevant items retrieved for a query image we use the Zap50K class information for the shoes. If the retrieved image class is same as the query image class than the result is marked relevant, and if the retrieved image class is different than of the query image it is marked as irrelevant.

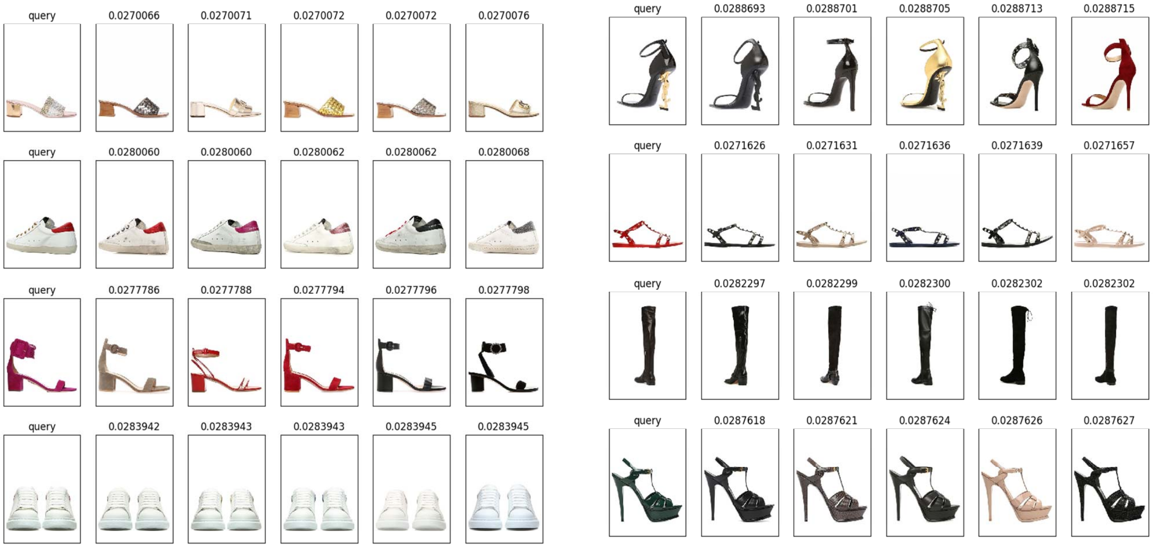

Fig.5. Sample visual similarity results of the proposed model on unseen images retrieved from beymen.com (Query image and top-5 similar images)

Performance results reported in Table 2 shows that, as the depth of the neural networks increases (such as VGG16 (16 Layer), VGG (19 Layer), Resnet 152 (152 Layer)), the additional cost of inference time increase which might lead to negative user-experience. On the other hand, our proposed network takes an average of 0,004 seconds per query image, which is significantly faster and also provides higher precision rates than other heavy pre-trained models. Fig.4. shows the similarity results for the proposed model and other models tested in this study for a sample shoe image. Fig.5. presents similarity results for randomly selected shoe images from unseen test dataset.

VI. Conclusion

This paper summarizes our work on developing a deep learning based image retrieval model for e-commerce sites. The proposed network in this study is based on InfoGAN with several changes in the network architecture for achieving better results for the similar shoe image retrieval problem. We compare several well-known architectures for shoe image similarity with the proposed network. The results show that the proposed model achieves superior performance in terms of precision and time. We conclude that the proposed model can be used in real-world e-commerce solutions since it can provide accurate and fast inference results. We plan to integrate the model with an e-commerce platform to provide better product recommendations for the customers.

References

[1] Smeulders, Arnold WM, et al. "Content-based image retrieval at the end of the early years." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 12 (2000): 1349-1380.

[2] Gudivada, Venkat N., and Vijay V. Raghavan. "Content based image retrieval systems." *Computer* 28.9 (1995): 18-22.

[3] Premachandran, Vittal, and Alan L. Yuille. "Unsupervised learning using generative adversarial training and clustering." (2016).

[4] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

[5] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

[6] Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[7] Zhang, Han, et al. "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

[8] Kim, Taeksoo, et al. "Learning to discover cross-domain relations with generative adversarial networks." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

[9] Li, Tingting, et al. "Beautygan: Instance-level facial makeup transfer with deep generative adversarial network." *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018.

[10] Creswell, Antonia, et al. "Generative adversarial networks: An overview." *IEEE Signal Processing Magazine* 35.1 (2018): 53-65.

[11] Hou, Xianxu, Ke Sun, and Guoping Qiu. "Deep Feature Similarity for Generative Adversarial Networks." *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2017.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[13] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv preprint arXiv:1511.06434, 2015.

[14] X. Hou, L. Shen, K. Sun, and G. Qiu, “Deep feature consistent variational autoencoder,” arXiv preprint arXiv:1610.00291, 2016.

[15] Fu, Jianlong, et al. "Efficient clothing retrieval with semantic-preserving visual phrases." Asian conference on computer vision. Springer, Berlin, Heidelberg, 2012.

[16] Liu, Qiang, Shu Wu, and Liang Wang. "Deepstyle: Learning user preferences for visual recommendation." Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2017.

[17] Wang, Xianwang, and Tong Zhang. "Clothes search in consumer photos via color matching and attribute learning." Proceedings of the 19th ACM international conference on Multimedia. ACM, 2011.

[18] Zhou, Zhengzhong, et al. "Interactive Image Search for Clothing Recommendation." Proceedings of the 24th ACM international conference on Multimedia. ACM, 2016.

[19] Liu, Si, et al. "Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set." 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012.

[20] Feng, Zunlei, et al. "Interpretable partitioned embedding for customized multi-item fashion outfit composition." Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval. ACM, 2018.

[21] Hadi Kiapour, M., et al. "Where to buy it: Matching street clothing photos in online shops." Proceedings of the IEEE international conference on computer vision. 2015.

[22] Huang, Junshi, et al. "Cross-domain image retrieval with a dual attribute-aware ranking network." Proceedings of the IEEE international conference on computer vision. 2015.

[23] Khosla, Neal, and Vignesh Venkataraman. "Building image-based shoe search using convolutional neural networks." CS231n course project reports (2015).

[24] Shankar, Devashish, et al. "Deep learning based large scale visual recommendation and search for e-commerce." arXiv preprint arXiv:1703.02344 (2017).

[25] Kota Yamaguchi. 2012. Parsing Clothing in Fashion Photographs. In Proc. CVPR. 570–3577.

[26] Yu, Aron, and Kristen Grauman. "Fine-grained visual comparisons with local learning." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[27] Chen, Xi, et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets." Advances in neural information processing systems. 2016.

[28] Iandola, Forrest, et al. "Densenet: Implementing efficient convnet descriptor pyramids." arXiv preprint arXiv:1404.1869 (2014).

[29] Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[30] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[31] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[32] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[33] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).

[34] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.