Clothing Retrieval and Visual Recommendation for Fashion Images

James Le - jl1165@rit.edu
Rochester Institute of Technology

Abstract

The fashion domain is a very popular playground for applications of machine learning and computer vision. The problems in this domain is challenging due to the high level of subjectivity and the semantic complexity of the features involved. Recent work has focused on a variety of approaches including attribute recognition, clothing retrieval, image generation, and visual recommendation. In this project, I developed a two-stage deep learning framework that can retrieve clothing images from the data and then visually recommend similar images for specific fashion styles. I trained my model on the Fashion144k dataset and tested it on the DeepFashion dataset. The experiments demonstrate the effectivess of my model.

1. Introduction

Fashion brands of all sizes and specialties are using technology to understand customers better than ever before. As those data collection efforts grow more sophisticated, artificial intelligence will reshape brands' approach to product design and development, with a focus on predicting what customers will want to wear next.

Stitch Fix [3] is already at the forefront of AI-driven fashion with its hybrid design garments, which are created by algorithms that identify trends and styles missing from the Stitch Fix inventory and suggest new designs for human designers' approval. The clients sign up to receive customized boxes of clothes composed for them, known as a Fix. The Fix is matched to a personal stylist by an algorithm based on the client's style preferences and then assigned to a specific warehouse based on another algorithm.

Rent the Runway [2] is another company that utilizes AI in the fashion realm. The company provides rentals of designer dresses to those who might otherwise not be able to afford them or may not need them long-term. Data analytics, recommendation system, and machine learning problems are just a few areas where AI is in use.

Pinterest [1] has also done some amazing computer vision work in visual search, where fashion is a popular use case. As seen in figure 1, the Pinterest Lens product allows users to discover ideas without having to find the right words to describe them first. Just point Lens at a pair of shoes, then tap to see related styles or even ideas for what else to wear them with. Or try it on a table to find similar designs, and even other furniture from the same era. Users can also use Lens with food. Just point it at cauliflower or potatoes to see what recipes come up. Patterns and colors can also lead users in to fun, interesting, or even just plain weird new directions.

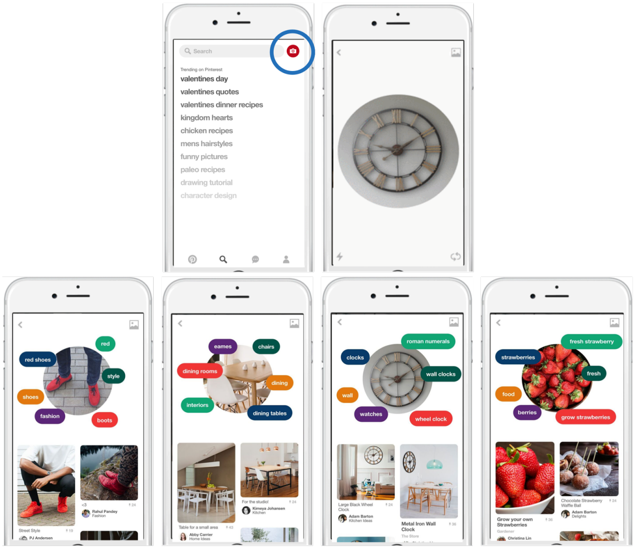

Figure 1: Sample use cases of Pinterest Lens.

Within the larger artificial intelligence realm, computer vision is an important area of focus for fashion products, because a user's buying decision is primarily influenced by the product's visual appearance. This brings me to the problem of **visual recommendation**, which incorporate visual signals directly into the recommendation objective. A user interested in buying a particular item from the screen may want to explore visually similar items before finishing her purchase. These could be items with similar colors, patterns, and shapes. Another interesting problem is that of **clothing retrieval**. This means to help users localize clothing items from regular pictures taken in an open environment so they can retrieve it.

The main contribution of this project is an end-to-end solution for large-scale clothing retrieval and visual recommendation. More specifically, my system can learn the important regions in an image and generate diverse recommendations based on such semantic similarity. This follows some of the recent works that incorporate visual features into content-based filtering recommendation system [10], [16], [19].

Methodologically, my system is heavily inspired by the work done by Wang et. al [37], which consists of 2 branch networks: a global branch based on Convolutional Neural Networks and an attention branch based on Visual Attention Model. Similarly, I will also use CNN to extract global image features and an attention module to learn which locations in the image are worthy of attention during training. I'll combine the intermediate feature maps and the attention maps to come up with the feature vectors for retrieval. Next, I'll rank these feature vectors using the simple k-Nearest Neighbor algorithm. The end result will be top-k images that are similar to the input image.

Quantitatively, I perform experiments on two popular fashion dataset named *Fashion144k* [29] and *DeepFashion* [22]. *Qualitatively*, my model recommends reasonable and diverse clothing items that can potentially matche the individual user's preferences. This aligns well with recent trends in developing recommendation systems to make them more nuanced and more diverse - being able to capture fine-grained textures and granularity of the visual items.

2. Related Work and Background

In this section, I first give an overview of the current fashion attributes recognition solutions proposed in the literature. Secondly, I discuss several approaches used in the context of image generation. I then present works related to clothing image retrieval, as it directly related to my proposed method. Finally, I present an overview of existing recommendation methods in fashion domain as this topic is gaining popularity.

2.1. Attribute Recognition

Different clothes have different attributes. For example, your shoes can be available in different colors and sizes, your shirts can be available different textures and patterns, and your pants can be available in different materials and width. A good understanding of such clothing attributes is extremely useful to do any sort of comparison between the clothing products. Due to limited computing power, works in 5-7 years back had been able to only use hand-crafted features such as HOG [5] and SIFT [23]. Thanks to the advance in deep neural networks recently, the computer vision research community has put a lot of efforts into building models that can recognize latent attributes from clothing images.

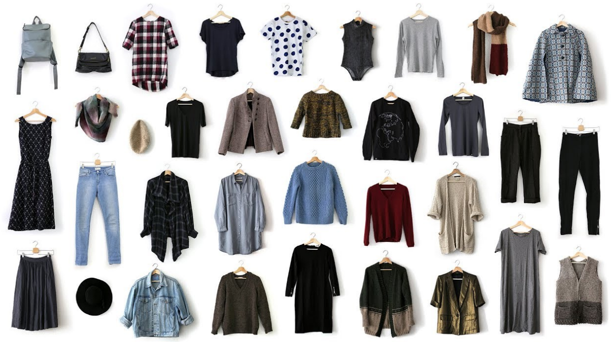

Figure 2: A sample capsule wardrobe. [12]

Li et al. [20] proposed a clothing recognition framework that is inspired by extreme learning machines (ELMs). Extreme learning machines [32] are simply feed-forward neural networks in which we do not need to manually tune the hyper-parameters of the hidden layers. This framework makes use of more advanced ELMs models known as Autoencoder ELMs and Ada-ELMs. Specifically, the framework first pr-eprocesses clothing images to extract 3 types of features (CNN, HOG, and Color Histogram), next concatenates these features using a simple feature-level fusion, and finally passes the fusion representation through the ELMs models to predict the clothing category.

Hsiao and Grauman [12] created a model that does exceptionally well in cross-domain attribute recognition to solve the capsule wardrobe problem. For those who are not familiar with the term, a capsule wardrobe is a small collection of essential items that can fit with a variety of outfits (see Figure 2). Specifically, their model first uses transfer learning with ResNet-50 [9] pretrained on ImageNet [6] to recognize the attribute of the clothing catalog. Then, they use object detection techniques to classify such attributes into top and bottom clothing outfits. Finally, they fine-tune the network to generate outfit compositions from these attributes. The model gets very good results for challenging attributes that can be beneficial to predict visual compatibility for each layer of clothing.

A new approach known as FashionSearchNet [4] is an interesting model that can search for fashion images using only query images and attributes. The network structure is very similar to that of the famous AlexNet [18], except that it removes all the fully-connected layers and uses 7 convolutional layers in total (AlexNet has 5 convolutional layers). The network is trained to generate Attribute Activation Maps (AAMs) [43], which then extracts the Regions of Interest (ROIs) that represent the most activated attribute regions. Lastly, the model combines all these attributes into a single global representation that generalizes well to most

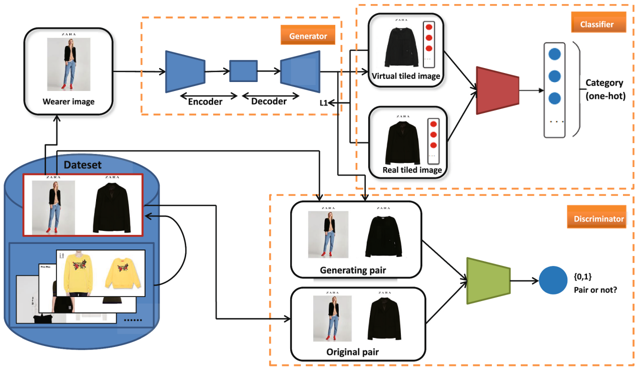

Figure 3: Overview of the ClothingOut model. [39]

fashion images.

2.2. Image Generation

Artificial intelligence can not only learn a piece of clothing's attributes, but also can create computer-generated images of similar-looking items. This is quite valuable, especially for retailers, to create personalized clothes or even predict broader fashion trends. Unfortunately, generating realistic-looking fashion images has been a challenging task due to their high-dimensions. In order to cope with that, researchers again look towards deep learning techniques. In particular, recent approaches in image generation have made heavy use of generative adversarial networks (GANs) [8], a popular unsupervised machine learning model where there are two neural networks fighting against each other.

Zhang et al. [39] came up with ClothingOut, a GAN-based model that can generate clothing images using just their semantic information. As you can see in Figure 3, the images are first put into a generator network that consists of an encoder-decoder module to generate fake images, then passed onto a discriminator network that compare the fake and original images, and finally given to a classifier network that can classify the categories for the fake images.

Ma et. al [24] also used GAN for their image generation framework that focuses on the person's poses. Known as the Pose Guided Person Generation Network, their model consists of two stages. In the first stage, their network integrates the poses in image using a pose estimator to get the human poses, a generator G1 to generate initial coarse results, and pose mask loss to properly weigh the human body compared to other background objects. In the second stage, their network refines the generated image using a generator G2 to make that image closer to the ground truth and a discriminator D to do the comparison.

Most recently, Zhao et. al [41] built a GAN-based model that can generate images in different views using only a single-view image. Known as VariGANs, the model is a mash-up of the variational auto-encoder and generative adversarial networks. The network structure includes 3 main components. First one is a Coarse Image Generator with encoder-decoder module and word embedding layer that learn the latent image representation to generate coarse images. Second one is a Fine Image Generator with similar structure as the first one, except with the addition of skip connections to helps with high-resolution image tensor. Third one is a Conditional Discriminator that distinguishes the realness and fakeness of the generated images.

2.3. Clothing Retrieval

Clothing retrieval is essentially a subset of image retrieval, an ongoing active research in computer vision domain. This technique attempts to identify the topic of an image, find the right keywords to index the image, and define the appropriate words to retrieve that image. There is a semantic gap in between these objectives, making the meaning of an image to be highly individual and subjective. With the large amount of image data, image retrieval on a big dataset becomes an even more challenging visual task.

A simple approach for fashion image retrieval proposed by Gajic and Baldrich [7] is a Siamese network architecture, that makes use of adversarial learning. In particular, they used 3 different Siamese streams, where each stream uses a convolutional layer, a max-pooling layer, a fully-connected layer, and L2 normalization. This simple approach proves to outperform previous benchmark results on very specific clothing retrieval problem across domains.

Wang et al. [37] approached the clothing retrieval problem with an end-to-end system known as Visual Attention Model. The model consists of a global branch and an attention branch. Every input image is fed into both branches at the same time. The global branch creates feature maps using the lower layers of the input image, while the attention branch does the same thing using an attention map from a pre-trained fully convolutional network. The results of these maps are feature vectors of the original image that can be used for retrieval. This network also utilizes Impdrop connection, a method similar to Dropout [11] that makes the model less prone to overfitting.

Another end-to-end method, known as DeepStyle, came from Taukute et al. [33]. Their overall search engine makes use of 4 different techniques: (1) Visual Search that uses ResNet-50 model to detect objects and extract features from the images, (2) Text Query Search that uses Continuous Bag-of-Words model to map the textual description embeddings to the image's vector representations, (3) Context Space Search that uses a word2vec model to retrieve more product data, and (4) Blending Methods that essentially merge all the retrieval results from the search above. Figure 4 displays the core structure of DeepStyle. The model

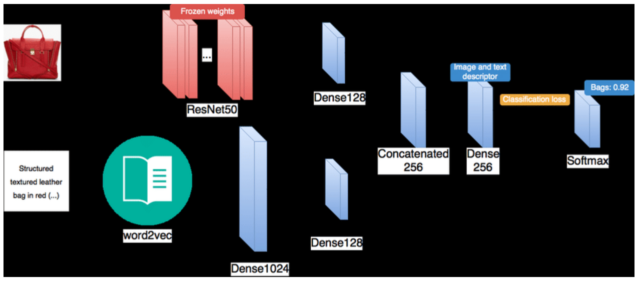

Figure 4: The main architecture of DeepStyle network. [33]

uses a pre-trained ResNet-50 model to deal with the image features of the product and a word2vec model to deal with the textual descriptions. The resulting features are then fed into a series of dense layers and concatenated into a single vector, which then is classified with a softmax prediction.

Perhaps the most interesting application of deep learning in clothing retrieval is Zhang et al.'s DeepLink [38], which attempts to retrieve clothing images of movie stars from sitcom TV shows. The working flow of DeepLink framework goes like this: The input video is treated as a sequence of separate images. For every pass, the model picks up one frame in a second as an input and passes the frame through a series of modules in this order - (1) Human Body Detection, (2) Pose Selection, (3) Face Detection and Verification, (4) Clothing Detection and (5) Clothing Retrieval. Each of these modules consists of very deep CNN architectures. The model has been proven to achieve competitive results compared to previous benchmark results.

2.4. Fashion Recommendation

Recommendation engines are becoming more common in the fashion area. Many current industry applications allow shoppers to scan clothes, bags, shoes, and other accessories to get a personalized suggestion. These engines focus on a variety of clothing attributes including colors, patterns, size, weight, and shape. However, as fashion is a very dynamic and ever-changing industry, there are many external factors that make building a fashion recommendation system challenging, such as public view, outfit guidelines, current trends etc.

Tuinhof et al. [34] tackled this problem using an image-based approach, in which only a single input image is needed to recommend a list of items ranked by similarity score. A fashion image is first put into a CNN classifier, where simple image classification techniques are performed to predict its category and texture type. The author used AlexNet [18] and batch-normalized Inception as their CNN classifiers for their experiments. Then, k-Nearest Neighbors algorithm is used to rank these features in a high-dimensional space. The end result is a list of top-k recommendations that are most similar to the input fashion image.

On the other hand, Liu et al. [21] approached fashion recommendation system using a quadruple network architecture with 4 stages. For stage 1, they adopted the Faster R-CNN [27] framework to detect the upper and lower-body regions of the input fashion image. For stage 2, they used a Siamese CNN [25] to learn the compatible space of these regions. In stage 3, they used a transformation CNN model based on AlexNet [18] to learn the upper and lower body style space of these regions. In stage 4, they extract the compatible and style features from stage 2 + 3 and fuse them together to achieve the final output visual features. They finally applied nearest neighbor algorithm to these features to retrieve the recommendation list for the input image.

Verma et al. [35] proposed a much deeper network architecture that can explicitly learn the discriminative features of the input image. Their neural network consists of 3 components: First, a CNN with 3 convolution layers and 3 max-pooling layers extract the global features of the input fashion image. Second, a Recurrent Neural Network (RNN)-based attention module focuses on the important regions within the feature vector space learned from the first stage. Third, a texture encoding layer captures the clothing texture of the attentions generated from the second stage. A huge advantage of this work is such that the recommendations are quite diverse as the similarity scores are calculated based on different parts in the input image.

The best commercial-focus recommendation system I have found so far is VisNet [28], which was built by the Flipkart team in India for the e-commerce space. VisNet's core architecture combines both a deep network and a shallow network. The deep network is a pretrained VGG-16 model without the final loss layer, while the shallow network is simply 2 convolution layers. The resulting features captured after these layers are then passed through L2 normalization and linear embedding layers to become a feature vector. Finally, k-nearest neighbor can be used to generate the similar images within this embedding feature space. The powerful capacity of VisNet is such that it can capture the high-level and low-level image details, which is highly beneficial to calculate the nuanced visual similarity.

3. Approach

My system has two important parts: a component for clothing retrieval (based on a combination of CNN, attention module, and texture encoder) and a component for similarity recommendation (based on the k-NN algorithm).

3.1. Clothing Retrieval

My proposed model for clothing retrieval consists of 2 branches: a global branch using convolutional neural network-based model to extract global image features and

type	kernel size	output size
convolution	3 x 3	384 x 256 x 64
convolution	3 x 3	384 x 256 x 64
dropout (25%)		384 x 256 x 64
max pooling	4 x 4	96 x 64 x 64
batch normalization		96 x 64 x 64
convolution	3 x 3	96 x 64 x 128
convolution	3 x 3	96 x 64 x 128
dropout (25%)		96 x 64 x 128
max pooling	4 x 4	24 x 16 x 128
batch normalization		24 x 16 x 128
convolution	3 x 3	24 x 16 x 256
convolution	3 x 3	24 x 16 x 256

Table 1: Global feature extraction network architecture. [30]

an attention branch using attention model to learn important semantic regions in the images. Such combination will make the model more stable and more robust towards disturbances in the images such as background and occlusions. Each image is fed into these branches at the same time. The resulting global feature map and attention map must have identical height and width. At the end, the final feature matrix of the input image is the concatenation of both global features and attention features.

Global Branch: To extract the global image features, I follow the approach used by Simo-Serra in [30], whose model has been trained specifically for clothing and fashion images. Table 1 is the overview of the network architecture used to extract global images. All convolutional layers have 1×1 padding and all layers besides the max pooling layers have a 1×1 stide, while all max pooling layers have a 4×4 stride. I use 3×3 kernels for the convolutional filters to reduce the number of weights and increase the number of layers in the network. I also use Batch Normalization layers [14] to maximize efficient learning and Dropout layers [11] to prevent overfitting.

Attention Branch: To learn important semantic regions in the images, I try a recurrent memorized-attention module as used by Wang in [36]. It combines the recurrent computation process of an Long-Short Term Memory network with a spatial transformer and a texture encoding layer. It iteratively searches the most discriminative regions and predicts the scores of label distribution for them as well as the corresponding feature vectors.

The spatial transformer module proposed in [15] is designed specifically to handle the spatial variation of the data, where pooling layers previously have shown some issues. It spatially transforms its input maps to the output maps with a given size that correspond to a sub-region of the input maps. Figure 5 shows an example of spatial transformation.

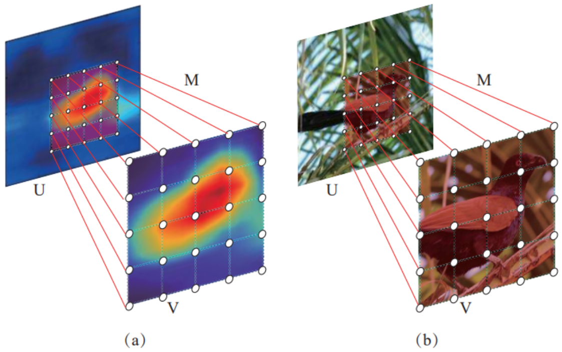

Figure 5: Illustration of spatial transformation on (a) the feature maps and (b) the corresponding input image. [36]

A transformation matrix is first estimated by a localization network. After that, the corresponding coordinate grid is obtained based on the coordinates of the attentional region. The, the sampled feature maps that correspond to the attentional region are generated by bilinear interpolation.

It is convenient to embed a spatial transformer layer in the neural network and train it with the standard backpropagation algorithm. In my model, I incorporate it in the recurrent memorized-attention module for the localization of attentional regions. It is also computationally fast and does not affect the training speed. More importantly, it performs active spatial transformation on a feature map for each input sample, as compared to the pooling layer which acted identically for all input samples.

The textual encoding layer proposed in [40] receives the features extracted from the spatial transformer and learns the texture of each attentional regions. This layer is designed specifically to represent spatial invariance of the feature distribution in an image. It has 3 very attractive properties. (1) It is an orderless pooling layer, which is suitable for material and texture recognition. (2) It accepts arbitrary input sizes, which makes the training process more flexible. Finally, (3) It generalizes dictionary learning and encoding framework (which carries domain-specific information), making it suitable for domain transfer learning. In this work, since the network is trained end-to-end, the convolutional features learned with Encoding Layer on top are easier to transfer.

3.2. Similarity Recommendation

A simple algorithm I use to rank features is **k-Nearest Neighbors** (k-NN). Essentially, the algorithms forms a majority vote between the K most similar instances to a given unseen data point. The similarity distance that I use is Euclidean distance given by this equation:

$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2} \tag{1}$$

Given a similarity distance d and an unseen data point a, while running through the whole dataset, kNN computes d between a and each training data point. The K points in the training data that are closest to a are grouped into a set (let's call it A). The algorithm then estimates the conditional probability for each class, which is the fraction of points in A with the given class label. It is given by the equation:

$$P(y = j | X = a) = \frac{1}{K} \sum_{i \in A} I(y_i = j) \tag{2}$$

Recall that the output of the clothing retrieval system is a feature matrix. I apply the k-NN ranking algorithm to this feature matrix and retrieve the top-k matching style recommendations with a given input clothing image.

4. Experiments

I implement my approach using the PyTorch framework [26]. I train my model on the Fashion144k dataset [29], and evaluate on the DeepFashion dataset [22]. I compare my results against baseline models that have been tested on DeepFashion.

4.1. Datasets

Fashion144k: I train my model on the Fashion144k dataset [29], which has been collected by the Waseda University in Japan. It contains roughly 90,000 training images of 128 classes with multi-label annotations as well as fashionability scores. Each image has 384 × 256 resolution. Figure 6 illustrates an image from the dataset. Each image contains text in the form of descriptions and garment tags, as well as comments from people. It also contains votes which are used as a proxy for fashionability.

This dataset is useful for training because of the availability of multiple labels (59 in total) and associated metadata that can be used for analysis and prediction. My model is trained with all 59 item labels, while color labels are excluded.

DeepFashion: I evaluate my model on the DeepFashion dataset [22], which has been collected by the Chinese Hong Kong University. It has over 800,000 diverse fashion images and rich annotations with additional information about landmarks, categories, pairs etc. Figure 7 displays some sample images from the dataset.

The dataset consists of 5 different kinds of predicting subsets that are tailored towards their specific tasks. One subset, called In-Shop Clothes Retrieval, can be used for recognition and retrieval on clothing images. With more than 52,000 in-shop images of nearly 8,000 clothing items, this subset is ideal for me to experiment on. The task is to figure out if two images taken in shop belong to the same clothing item. This is important because when customers see a shop image online, they most likely would like to learn more about its item information on online retailer stores.

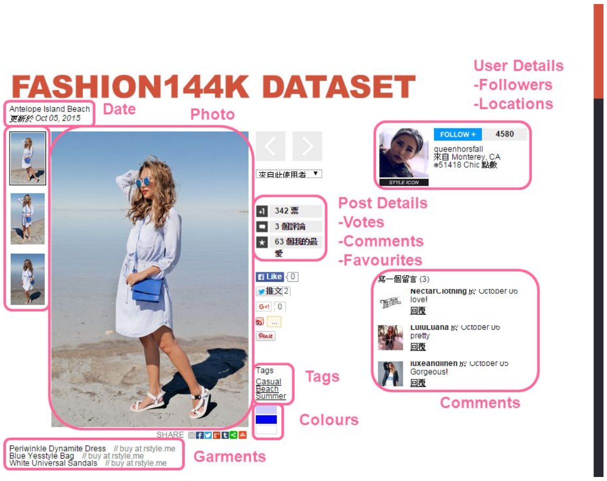

Figure 6: Anatomy of an image from the Fashion144k dataset. [29]

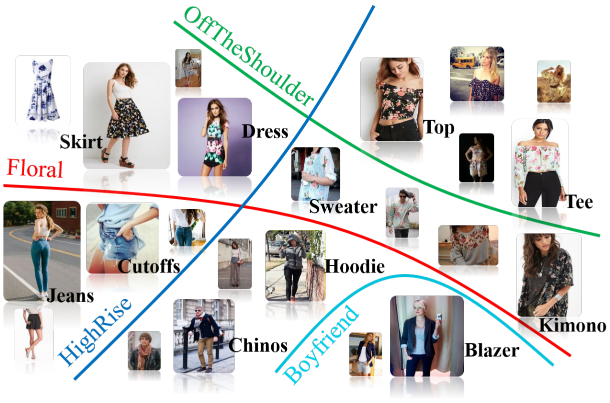

Figure 7: Sample images from the DeepFashion dataset. [22]

4.2. Training

During backpropagation, I use 3 different kinds of loss functions, including a classification loss, a diversity loss, and a localization loss. First, to handle multi-label image classification task on Fashion144k, I use the **multi-label classification loss** given in [36], which employs the Euclidean loss as the objective function and can be expressed as:

$$L_{cls} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}(p_i^c - \hat{p}_i^c)^2, \tag{3}$$

where N is the number of training samples, C is the number of class labels, $\hat{p}_i$ is the ground-truth probability vector of the i-th sample, and p_i is the predicted label vector of that sample i.

Second, to diversity the attention regions, I use the **diversity loss** given in [42] to compute the correlation between temporally adjacent attention maps:

$$L_{div} = \frac{1}{K-1}\sum_{k=2}^{K}\sum_{i=1}^{H\times W} l_{k-1}, i.l_{k,i}, \tag{4}$$

where K is the total steps of recurrent attention, $H \times W$ is the height and width of attention maps, l_k is the k-th attention map, and $l_{k,i}$ is the i-th attention value of the attention map after conducting softmax on $H \times W$ locations at time step t.

Third, to remove redundant locations and force localization network to look at small clothing parts, I use the **localization loss** given in [36], which consists of three constraints on the parameters of a transformation matrix:

Anchor constraint: This constraint pushes the attentional regions away from the image center, which helps scatter these regions over different semantic parts in the image. It can be formulated as:

$$L_A = \frac{1}{2}\{(t_x^k - c_x^k)^2 + (t_y^k - c_y^k)^2\}, \tag{5}$$

where (c_x^k, c_y^k) is the location of the k-th anchor point.

Scale constraint: This constraint pushes the scale parameters in a certain range, so that the located attentional region won't be too large. It can be formulated as:

$$L_S = L_{s_x} + L_{s_y}, \tag{6}$$

in which

$$L_{s_x} = (max(|s_x| - \alpha, 0))^2$$

$$L_{s_y} = (max(|s_y| - \alpha, 0))^2$$

where α is a threshold value, and it is set as 0.5 for my experiments.

Positive constraint: This constraint is used to restrict the scale parameters. It prefers a transformation matrix with positive scale parameters, leading to attentional regions that are not mirrored:

$$L_P = max(0, \beta - s_x) + max(0, \beta - s_y), \tag{7}$$

where β is a threshold value, set as 0.1 for my experiments.

Finally, I combine the 3 constraint equations in (5), (6), (7) to define the localization loss. It is formulated as the weighted sum of the 3 components:

$$L_{loc} = L_S + \lambda_1 L_A + \lambda_2 L_P, \tag{8}$$

where λ_1 and λ_2 are the weighted parameters, and they are set as 0.01 and 0.1, respectively.

My model is jointly trained with the classification loss, the diversity loss, and the localization loss, so the overall combined loss function can be expressed as:

$$L = L_{cls} + \gamma_1 L_{div} + \gamma_2 L_{loc}, \tag{9}$$

where γ_1 and γ_2 are multiplicative factors, and they are set as 0.01 for my experiments.

During training, I minimize the combined loss function with Adam optimizer using batch size of 64, learning rate of 0.00001, and momentum of 0.9. The model is trained over 40 epochs, which is when the loss converges. I save the resulting weight from this training process, so I can reuse it during the testing process.

4.3. Quantitative Evaluation

Baselines: For the clothing retrieval task, I compare my method against the following baselines:

- **FashionNet:** Introduced in the DeepFashion paper [22], this model has a similar network structure to that of VGG-16 [31], except that the last convolutional layer has been designed specifically for clothes.
- **Where To Buy It (WTBI):** Proposed in [17], this model is trained with bounding boxes and uses a multi-layer perceptron on top of VGG-16 [31] pretrained on ImageNet.
- **Dual Attribute-aware Ranking Network (DARN):** Proposed in [13], this model is trained with bounding boxes and uses two streams of CNN to regularize the clothing attributes - one for shop images and the other for street images.
- **VAM**: Proposed in [37], this model uses a self-learning visual attention map coupled with Impdrop connection (similar to Dropout). It is also the current state-of-the-art on DeepFashion In-shop Retrieval Benchmark.

Evaluation Protocol: For clothing retrieval performance, I use top-k retrieval accuracy, jut as in [22] and [17]. It is adopted to measure the performance of fashion retrieval - a successful retrieval means that an exact fashion item has been found in the top-k retrieved results. The top-k

Method	Top-5	Top-10	Top-20	Top-30	Top-50
FashionNet	0.678	0.725	0.764	0.781	0.796
WTBI	0.425	0.470	0.506	0.514	0.540
DARN	0.548	0.624	0.675	0.701	0.719
VAM	0.836	0.887	0.923	0.936	0.947
Mine	**0.683**	**0.728**	**0.775**	**0.802**	**0.834**

Table 2: Retrieval accuracies of the compared methods.

images are calculated to be the nearest to the input image using Euclidean distance. For recommendation performance, a precise metric to evaluate recommendation quality is difficult because of the subjective nature of recommendation system.

Results: Table 2 shows the top-k retrieval accuracy of all the compared methods with $k = 5, 10, 20, 30, 50$. The results show that my model achieves the second best performance out of the 5 methods, lagging behind only the VAM approach. It is not too surprising that WTBI performs poorly since it uses a network pre-trained on ImageNet dataset that is not suitable for clothing features. Most notably, my model slightly edges out FashionNet, the benchmark architecture used on DeepFashion, with about 1% improvement. This reveals the merits of learning to attend to different parts as well as employing texture-based features to find clothing similarity.

4.4. Qualitative Evaluation

For the recommendation task, I extract the features from an input image, submit them to the k-NN ranking algorithm and return the top-k matching style recommendations. In figure 8, I present several query images and corresponding top-5 recommendations. The query images are marked in black, going from top row down: women's rompers, men's jackets, women's blouse shirts, men's hoodies, women's dresses, and men's sweaters. Subjectively, the top-5 recommendations indeed look quite similar to the query images. Indeed, I highlight the items that are correct matches of itself in green. This demonstrates that a clothing item is actually most similar to itself. A precise objective evaluation criterion for recommendation quality is difficult because of the inherent subjectivity of recommendation system. This makes comparison to other methods quite challenging.

5. Conclusion

In this work, I have presented a system for fashion recommendation that is capable of producing diverse recommendations on the basis of similarity of different parts in the query image. This suggests a new type of recommendation approach that can be used for both prediction and design. The proposed two-stage approach uses a CNN classifier to extract global features and a visual attention module to attend to semantic / texture-based features, which are used as input for similarity recommendations. To demonstrate the advantages of such components, I trained my model on Fashion144k and evaluated the results on DeepFashion. Through experiments, I show the effectiveness of my model, which may significantly facilitate future research.

References

[1] Get a new lens on life — about pinterest. https://about.pinterest.com/en/lens, Nov. 2018.

[2] Rent the runway - rent designer dresses, apparel and accessories. https://www.renttherunway.com/, Nov. 2018.

[3] Stitch fix: Fashion stylist — clothes boxes. https://www.stitchfix.com/, Nov. 2018.

[4] K. E. Ak, A. Kassim, J. H. Lim, and J. Y. Tham. Learning attribute representations with localization for flexible fashion search. In *CVPR, pages 7708-7717*, 2018.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR, pages 5315-5324*, 2015.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] B. Gajic and R. Baldrich. Cross-domain image retrieval with attention modeling. In *CVPR, pages 1982-1984*, 2017.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR, pages 770-778*, 2016.

[10] R. He and J. McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. In *AAAI*, 2016.

[11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. In *Computer Science, Volume 3, Issue 4, pages 212-223*, 2012.

[12] W.-L. Hsiao and K. Grauman. Creating capsule wardrobes from fashion images. In *CVPR*, 2018.

[13] J. Huang, R. S. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV, pages 1-6*, 2015.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariance shift. In *ICML*, 2015.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS, pages 2017-2025*, 2015.

[16] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley. Visually-aware fashion recommendation and design with generative image models. In *CVPR*, 2017.

[17] 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 *ICCV, pages 1-6*, 2015.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In

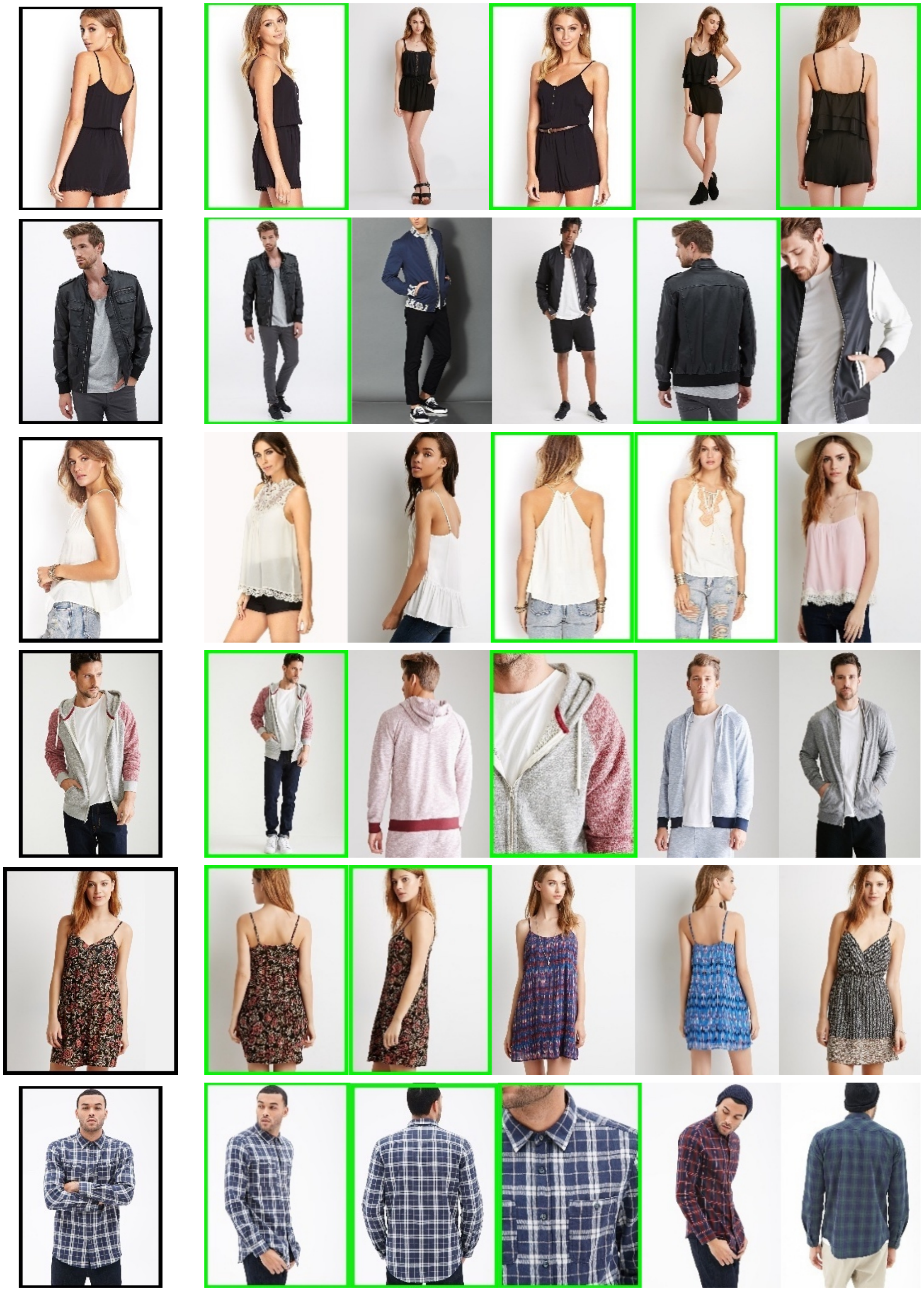

Figure 8: k-NN recommendation ranking. First colum displays the query images and columns 2-6 displays the predicted 5 nearest neighbors. Correct matches to the query image are marked in green.

Advances in neural information processing systems, pages 10971105, 2012.

[19] C. Lei, D. Liu, W. Li, Z.-J. Zha, and H. Li. Comparative deep learning of hybrid representations for image recommendations. In *CVPR*, 2016.

[20] R. Li, W. Lu, H. Liang, Y. Mao, and X. Wang. Multiple features with extreme learning machines for clothing image recognition. In *IEEE Accesss, Volume 6, pages 36283-36294*, 2018.

[21] Y.-J. Liu, Y.-B. Gao, Y.-L. Bian, W.-Y. Wang, and Z.-M. Li. How to wear beautifully? clothing pair recommendation. In *Journal of Computer Science and Tech, Volume 33, Issue 3, pages 522-530*, 2018.

[22] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] D. G. Lowe. Distinctive image features from scale-invarient keypoints. In *IJCV, Volume 60, Issue 2, pages 91-110*, 2004.

[24] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. V. Gool. Pose guided person image generation. In *NIPS*, 2017.

[25] I. Melekhov, J. Kannala, and E. Rahtu. Siamese network features for image matching. In *Proceedings International Conference on Pattern Recognition, pages 378-383*, 2017.

[26] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 39, Issue 6, pages 1137-1149*, 2017.

[28] D. Shankar, S. Narumanchi, H. A. Ananya, P. Kompalli, and K. Chaudhury. Deep learning based large scale visual recommendation and search for e-commerce. In *CVPR*, 2017.

[29] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. Neuroaesthetics in Fashion: Modeling the Perception of Fashionability. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[30] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[32] J. Tang, C. Deng, and G.-B. Huang. Extreme learning machine for multilayer perceptron. In *IEEE Transaction Neural Network Learning System, Volume 27, Issue 4, pages 809-821*, 2016.

[33] I. Tautkute, T. Trzcinski, A. Skorupa, L. Brocki, and K. Marasek. Deepstyle: Multimodal search engine for fashion and interior design. In *IEEE Transactions on Multimedia*, 2018.

[34] H. Tuinhof, C. Pirker, and M. Haltmeier. Image based fashion product recommendation with deep learning. In *CVPR*, 2018.

[35] S. Verma, S. Anand, C. Arora, and A. Rai. Diversity in fashion recommendation using semantic parsing. In *IEEE*, 2018.

[36] Z. Wang, T. Chen, G. Li, R. Xu, and L. Xin. Multi-label image recognition by recurrently disovering attentional regions. In *ICCV*, 2017.

[37] Z. Wang, Y. Gu, Y. Zhang, J. Zhou, and X. Gu. Clothing retrieval with visual attention model. In *IEEE*, 2017.

[38] H. Zhang, Y. Ji, W. Huang, and L. Liu. Sitcom-star-based clothing retrieval for video advertising: a deep learning framework. In *Neural Computing and Applications*, 2018.

[39] H. Zhang, Y. Sun, L. Liu, X. Wang, L. Li, and W. Liu. Clothingout: a category-supervised gan model for clothing segmentation and retrieval. In *The Natural Computing Applications Forum*, 2018.

[40] H. Zhang, J. Xue, and K. Dana. Deep ten: Texture encoding network. In *CVPR*, 2017.

[41] B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, Z. Jie, and J. Feng. Multi-view image generation from a single-view. In *CVPR*, 2018.

[42] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan. Diversified Visual Attention Networks for Fine-Grained Object Classification. In *IEEE Transactions on Multimedia*, 2017.

[43] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR, pages 29212929*, 2016.