International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Complementary Fashion Recommendation using Compatible Modelling

Shruthi S K[1], Shwetha S K[2], Adithi Shankar[3], Kruthika R[4], Deepthi Das V[5], Archana VR[6]

[1,2,3,4]Student, Department of AIML, Jyothy Institute of Technology, Bengaluru, India
[5,6]Assistant Professor, Department of AIML, Jyothy Institute of Technology, Bengaluru, India

Abstract
The last two years of COVID-19, repeated lockdowns, and plain fear of stepping out of the house have fast-paced the adoption of eCommerce by many decades. Total online spending in May 2020 hit $82.1 billion (about $210 per person in the US), up 77% year-over-year. Even though it is good news for retailers as they do not need to invest in numerous brick-and-mortar stores, they face the conundrum of serving an invisible customer. They cannot see the customer, do not have any insight into customer preferences, and might have a limited transaction history for the customers as well as the product sales. Walking into a store, salespeople see the customer and can recommend clothes based on the customer's looks. Our motivation is to create a cold start recommendation engine that does not need any knowledge of user preferences, user history, item propensity, or any other data to recommend products to the customer. In this paper we are trying to solve two problems, given a product, what are the similar products and what are the complementary products that will complete the outfit.

INTRODUCTION

Fashion retailers use sophisticated recommendation algorithms built on massive transaction volumes and search histories to both expressly and implicitly assist customers. Numerous cutting-edge algorithms suggest things bought by one user to other users that are similar to them based on user similarities. Amazon created the item-to-item collaborative filtering approach to recommend products based on item similarity when user data and transaction history became more limited.

Manual or machine-based image annotations, where tags for product category, color, and style are established, can also be used to create an item similarity vector.

Convolutional neural network (CNN) based fashion recommendation systems have received a lot of attention lately. These techniques suggest matching clothes to the customer automatically.

A Style feature decomposition method has been presented by Shin et al to extract the image style vector from the query image. However, because the clothes vector produced by this method combines information from both category and style, it is not useful for predicting compatibility between items from different categories.

LITERATURE SURVEY

One noteworthy study, "DeepStyle: A Clothing Style Embedding Network for Fashion Recommendation," by Zhang et al. (2023), proposes DeepStyle, a novel clothing style embedding network tailored for fashion

recommendation. The model utilizes a combination of CNNs and attention mechanisms to extract informative style features from fashion images, enabling accurate representation of clothing styles. By incorporating both global and local features, DeepStyle captures nuanced style attributes, facilitating more effective recommendations of visually appealing outfits to users.

Another significant contribution comes from "FashionGAN: Fashion Style Transfer with Adversarial Networks," by Wang et al. (2024), which introduces FashionGAN, a GAN-based framework for fashion style transfer. By training on a large dataset of fashion images, FashionGAN learns to encode diverse style variations and transfer them between clothing items. This capability enables the generation of personalized fashion recommendations by adapting clothing styles to match users' preferences and aesthetics, thereby enhancing user satisfaction and engagement.

Furthermore, "FashionBERT: Leveraging BERT for Fashion Recommendation with Textual and Visual Embeddings," by Chen et al. (2024), presents FashionBERT, a novel recommendation model that combines textual and visual embeddings for fashion items. By fine-tuning the BERT language model on fashion-related text data and incorporating image style embeddings obtained from pre-trained CNNs, FashionBERT achieves state-of-the-art performance in recommending fashion items based on their descriptions and visual attributes. This approach enables a more comprehensive understanding of fashion preferences and enhances the quality of recommendations for users.

Lastly, "StyleEmbed: Learning Fashion Embeddings with Self-Supervised Contrastive Learning," by Liu et al. (2024), introduces StyleEmbed, a self-supervised contrastive learning framework for fashion embeddings. By leveraging a large-scale dataset of fashion images, StyleEmbed learns to encode rich style representations in an unsupervised manner, capturing both global fashion trends and fine-grained style details. This learned embedding space facilitates effective retrieval of visually similar fashion items and enhances the diversity of recommendations, thereby improving user engagement and satisfaction in fashion e-commerce platforms.

PROPOSED METHODOLOGY

Similar Product Modelling

Image embeddings

The batch data was run using ResNet"18 to extract picture embeddings, and features were taken out of each image's final average pooling layer. Each batch's V embeddings were added to the final aggregated feature layer, which was pickled and saved in a style embedding catalog.

Similarity and Recommendation

The cosine distance function was utilized to obtain comparable products. Similar photos have a larger cosine similarity value, and the cosine function is easy to understand. The pairwise cosine similarity between the query feature vector and each embedding in the style embedding catalog is computed and sorted in descending order using the feature vector of the query product as input.

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Recommend complementary Products

The algorithm will forecast the products from different categories that are compatible with the original query product given an image of the query product. What kind of shoes or blouse would go with a pair

of leather pants, for instance, if that's the query product? In an ideal environment, the underlying subjectivity and individual preferences to couple different products lead to numerous X solutions to this problem.

IMPLEMENTATION

Dataset and Preprocessing

The training was consequently restricted to "37,890 outfits, which translates to single products." In the data, the ratio of female to male clothes is around 0:1.

Using the bounding boxes found in the data, individual photos were taken out for various polyvores. The sample technique was applied to generate an image triplet dataset. Each triplet comprises an anchor image (1), a positive image (2) selected from a different Polyvore category, and a negative image (3) randomly selected from a different image that belongs to the same category as the positive image

Compatible Product Modelling

The compatible product model is based on ResNet"18 with two additional FC layers on top with a batch normalization layer in between and a dropout layer, which outputs a set of "28-dimensional style embeddings for all products. Experiments showed that batch normalization was essential to the training process as it prevented the model from getting trapped into a trivial solution - making the embedding of anchor, positive, and negative embeddings identical, resulting in an extremely low loss. Making all embeddings identical would result in the pairwise compatibility score of " for all product pairs, causing failures in recommending any meaningful compatible products.

With the compatible model trained, all products were fed through the trained model to obtain the style embedding catalog, in which a pairwise compatibility score could be calculated, and recommendations could be performed.

The general process of the compatible modeling is shown below in Figure X and Figure I.

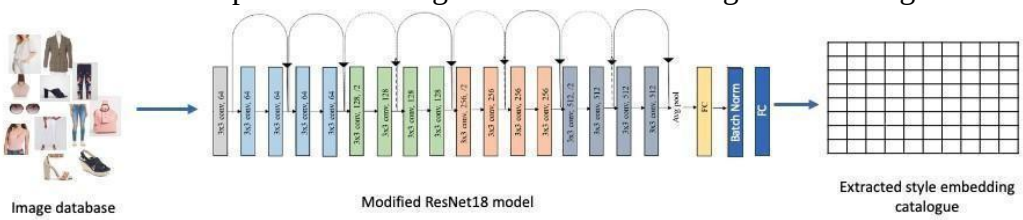

Figure 0— Flow of data as seen through the model

Figure 1— Model training to learn the compatibility of images

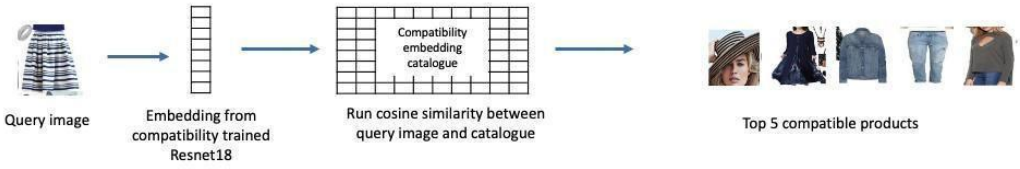

Figure 2— Extracting images compatible to the query image

TRAINING AND HYPERPARAMETERS TUNING

The model was trained with Triplet Loss, which is described as the following,

$$\mathcal{L}(A, P, N) = \max\left(\|\mathrm{f}(A) - \mathrm{f}(P)\|^2 - \|\mathrm{f}(A) - \mathrm{f}(N)\|^2 + \alpha, 0\right)$$

where A is embedding for the anchor image, P is the embedding for the positive image and N is the embedding for the negative image. The goal is that the distance from the anchor to the positive is minimized, and the distance from the anchor to the negative input is maximized.

80% of the total 77,393 triplets in the training data were used as training samples, while F% were saved for validation in the process. Each triplet was trained using the same network to acquire its feature embeddings, and the triplet loss algorithm mentioned above was used to compute the loss.

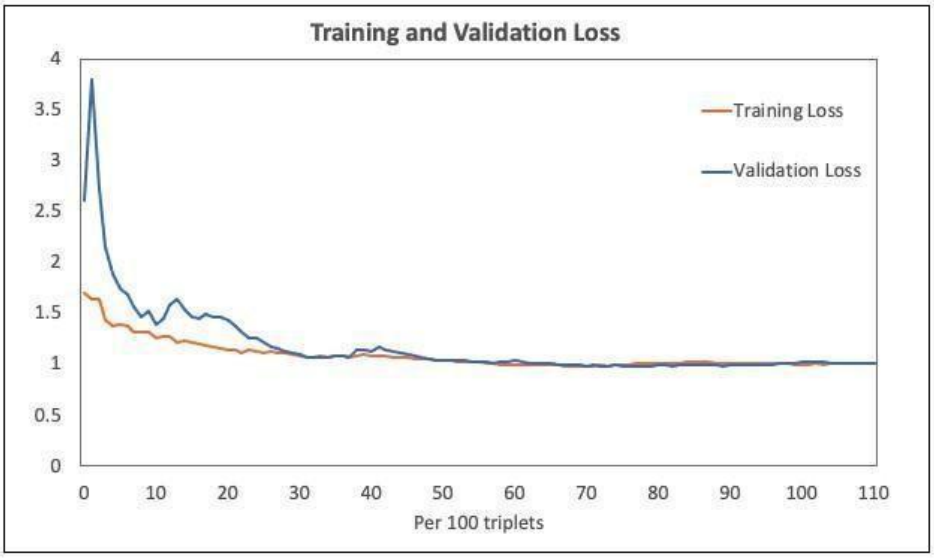

Figure 3— Loss curve for training the compatible model

To alleviate the bottleneck on the data loader, the batch size was set to 20. It was found that reading in the photos in larger batches accelerated the GPU-based training process, but would result in less than ideal performance when learning the style embedding representation. To enable speedier convergence, the triplet loss margin of 0.2 was selected. Higher margin values were leading to inconsistent losses.

EVALUATION

The test data, which included basic photos of IFEd outfits and produced EI, "X" only product items, was reserved for the final analysis. The process outlined in section E.E.V was used to create the style embedding catalog. A cropped image was selected from an outfit (anchor) and another cropped product was selected from the same image (positive) for every outfit base image. This combination was contrasted with a combination of an anchor and a cropped product from a different ensemble that belonged to the same category as the positive (negative) product picture.

In that record, the model is considered valid when the pairwise similarity score between the positive and anchor is greater than the similarity score between the negative and anchor. In particular, the accurate data from IFEd outfits were within the EHFF range (based on the sample seeds), resulting in a ~%\F accuracy rate. Figure H and Figure D show this comparison.

Figure +— This triplet shows a correct identification in compatibility

CHALLENGES

There were several challenges throughout the implementations.

The default shuffling behavior of Data Loader - The recommender returned incorrect picture metadata even if the feature vectors were accurately retrieved since the PyTorch data loader's default behavior was to randomly shuffle the training data. This led to a mismatch between the image's feature vector and its metadata. One way to solve this was to train the compatible model with shuffling set to True and extract the style embedding catalog with shuffling turned off.

Data Loader's Bottleneck: Even with GPU support, loading a triplet of photos has severely slowed down training speed; our team's best result was N hours for one epoch. With such a long training period, it was especially challenging to assess the performance of the suitable model or adjust the hyperparameter. A wonderful set of hyperparameters that would result in improved performances was difficult to isolate because there were very few resources available to solve the problem.

Compute Power: Deep learning on images requires a great deal of computing power. For this work, local compute resources on individual laptops were insufficient, so feature extraction and training on Google Cloud Platform (GCP) were utilized for a variety of compute-intensive tasks. Even with GCP, the model's ability to train for longer periods for more accurate results is severely limited because only E epochs could be run at a time to limit usage costs.

EXPERIMENTAL SETUP AND RESULTS

Recommend similar image results

Finding goods in the same product category that resembled the query image was the aim. For instance, if the user submitted a query image of a light-colored, knee-length coat, the results should include items that are stylistically similar and appealing to the user. Figure "F" displays some of the first findings from the "Recommend Similar Images" framework; more, in-depth results are displayed in the appendix in Figures "2, 3, and 5. Although evaluating a recommendation's quality is personal to the individual reading it, the initial evaluation of "00+ recommendations" indicates a high degree of confidence in the accuracy of comparable recommendations. The I suggested jackets resemble the original query image, but they are not exact, as Figure "F" illustrates.

Figure 5— The images on the left are query images and the images on the right are the ones returned by the "Recommend Similar Images" framework

Recommend complementary image results

The objective was to obtain goods that, while not belonging to the same product category, are stylistically compatible with the query image. Figure " displays some initial findings, and the appendix's figures "I, "\, and "N" provide more thorough results. As with the earlier method, human interpretation is included in the recommendation quality review process. It is safe to say that after looking over the "FF+ recommendations," the products that are suggested match the initial query image in terms of style. The suggested photos are a leather jacket, sneakers, a funky-looking sweatshirt, a fur hat, and a watch; all of these appear to be comparable to the initial query image, which depicts a black bag in figure ".

Figure 6— The images on the left are query images and the images on the right are the ones returned by "Recommend Complementary Images"

CONCLUSION AND FUTURE ENHANCEMENT

Recommend Similar Images and Recommend Complementary Images, both frameworks return exceptionally satisfactory results. From visual inspection, it is apparent that the experiment was successful. This paper aims to enable fashion retailers to create a recommendation system without the need for any customer history, customer preference, product sale history, product annotations, or any other such large-scale metadata. As future work, firstly, some experiments can be done with the Triplet Loss function such as soft triplet loss to further enhance the complementary recommendations. Secondly, an experiment with models other than ResNet-"18 as its base model to extract the embeddings could prove useful in showing improvements. Thirdly, for a true commercial representation, a combination of the approach recommended in this paper layered with transaction data insights to generate truly applicable recommendations. Finally, in the future, the application can be enhanced to where users can upload images of their clothes and the application would recommend the complement style or similar style.

REFERENCES

1. https://www.forbes.com/sites/johnkoetsier/2020/06/12/covid-19-accelerated-e-commerce- growth-4-to-6-years/?sh-6a4adc1f60of
2. Z. Liu, P. Luo, S. Qiu, X. Wang and X. Tang, "DeepFashion: Powering Robust Clothes Recog- nition and Retrieval with Rich Annotations," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1096-1104, doi: 10.1109/CVPR.2016.124.
3. Y. Shin, Y. Yeo, M. Sagong, S. Ji and S. Ko, "Deep Fashion Recommendation System with Style Feature Decomposition," 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), 2019, pp. 301-305, doi: 10.1109/ICCE-Berlin47944.2019.8966228.
4. Eileen Li, Eric Kim, Andrew Zhai, Josh Beal, Kunlong Gu, "Bootstrapping Complete the Look at Pinterest", arXiv:2006.10792, June 2020
5. Mahamudul Hasan, Shibbir Ahmed, Md. Ariful Islam Malik, and Shabbir Ahmed, "A com- prehensive approach towards user-based collaborative filtering recommender system," DOI:10.1109/IWCI.2016.7860358December 2016

6. Greg Linden, Brent Smith, and Jeremy York, "Recommendations Item-to-Item Collaborative Filtering," Amazon.com Industry Report 2003
7. Shuai Zheng, Fan Yang, M. Hadi Kiapour, Robinson Piramuthu, "ModaNet: A Large-Scale Street Fashion Dataset with Polygon Annotations," arXiv:1807.01394, July 2018
8. Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, Serge Belongie, "Learn- ing Visual Clothing Style with Heterogeneous Dyadic Co-occurrences," 1509.0747301 [CS.CV] 24 Sep 2015
9. Wang-Cheng Kang, Eric Kim, Jure Leskovec, Charles Rosenberg, Julian McAuley, "Complete the Look: Scene-based Complementary Product Recommendation," arXiv:1812.01748, April 2019
10. Ruining He, Charles Packer, and Julian McAuley, "Learning Compatibility Across Catego- ries for Heterogeneous Item Recommendation," arXiv:1603.09473, September 2016
11. Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, Rong Jin, "SoftTriple Loss: Deep Metric Learning Without Triplet Sampling," arXiv:1909.05235V2 [cs.CV] 15 Apr 2020
12. Tong He, Yang Hu, "FashionNet: Personalized Outfit Recommendation with Deep Neural Network," arXiv:1810.02443, October 2018
13. Chakraborty, S.; Hoque, M.S.; Rahman Jeem, N.; Biswas, M.C.; Bardhan, D.; Lobaton, E., "Fashion Recommendation Systems, Models and Methods: A Review," Informatics 2021, 8, 49. https://doi.org/10.3390/informatics8030049
14. Leon Gatys, Alexander S Ecker, and Matthias Bethge, "Texture synthesis using convolu- tional neural networks," In Advances in neural information processing systems, pages 262-270, 2015.