# Visual Recommendation on Pinterest Fashion Dataset


## Abstract

Predicting Fashion Compatibility is a challenging task due to its inherent complexity. Many of the existing works ignore the scene images and use only the product images to measure the similarity using these models for downstream recommendation tasks. In this work, we study the task called Shop the Look where given a scene image and product images that appear in the scene image, the model recommends visually compatible products. We use the scene images as anchor images and train the model using a triplet loss function which forces the model to learn meaningful embeddings that capture the semantics of the data. We present the results of our experiments using our model and other baselines.


## Introduction

Fashion recommendation is one of the hot topics for e-commerce. Most of the research done till now is on the compatibility of the a particular fashion and recommending other compatible fashion items. However, this results in loss of useful information about the user.

In this work, we deal with the Shop the Look task where we have a pair of scene image and query object along with its bounding boxes We use Pinterest Fashion Compatibility Dataset which provides us with JSON formattted files containing the hash ids of the corresponding products and scenes along with the bounding box if the product. Using the hash ids of the product and the scene, we obtain the images for the product and scene respectively using the APIs provided by Pinterest.

We analyse the number of products belonging to each category mentioned in the figure below.

We also analyse the number of products belonging to each sub category.

## Exploratory Data Analysis

As shown in figure 3, Shop the Look datasets consist of scene-product pairs, where the task is to retrieve and recommend visually compatible products based on the given scene. This scenario can be likened to the real world case where a user uploads an image of a celebrity and is recommended with visually compatible items.

In this work, we have used the Shop the Look Dataset from Pinterest [3]. This data consists of scene-product

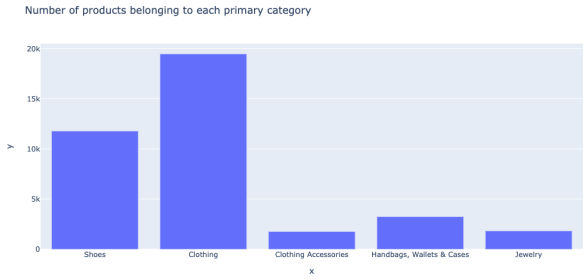


Figure 1: Distribution of data w.r.t. to coarse grained categories

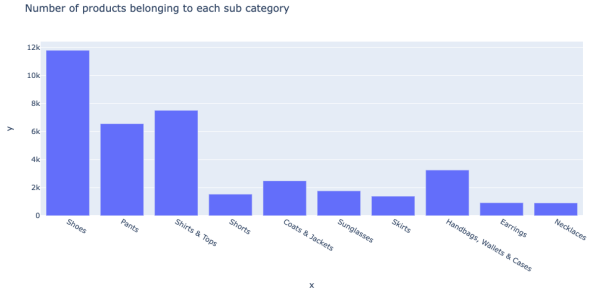


Figure 2: Distribution of data w.r.t. fine grained categories

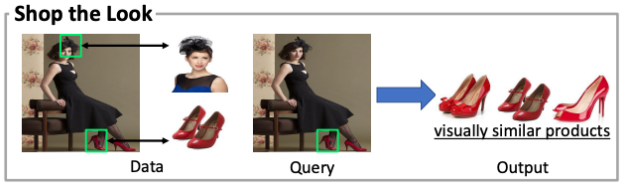


Figure 3: STL dataset

Table 1: Dataset Statistics

| Name | scene | product | pair | Product Categories |
|---|---|---|---|---|
| STL-Fashion(Pinterest) | 47,739 | 38,111 | 72,198 | shoes, tops, pants, handbags, coats, sunglasses, shorts, skirts, earrings, necklaces |


Figure 4: Products and their categories as present in the STL dataset

pairs. The scene images contains images of people who have bought certain items along with the bounding boxes of the products that they have bought that appear in the image and the product data consists of images of all products that appear in any of the scenes. The pinterest data is composed of STL-Fashion and STL-Home, where STL-Home contains images of interior design and home decor items. In this study, we have used only the STL-Fashion data. The statistics of the data are given in table 1.

Fig. 4 shows some of the products from our dataset. Fig. 5 shows sample scenes from our STL-Fashion data. We have multiple products available for the same scene in some cases. Apart from STL dataset, we also found a dataset called Exact Street2Shop as introduced in [1]. This data was collected from ModCloth where people could add photos of them wearing various products thus showing the exact items that they wear. We have not used this data in our analysis because we found the Pinterest data to be much larger and hence better suited for analysis.

We have made use of visual features to form the architecture of our model mentioned in the architecture section.

(a) (b)

Figure 5: Anchor Images with their corresponding bounding boxes on shoes and pants

## Predictive Task

We recommend products compatible to the product that appears in the scene. We call this task as Shop the Look. We evaluate using the Top K accuracy where the model will be deemed correct if the ground truth exists amongst the top K predictions. This way we will be able to understand if our model is predicting visually compatible products or not.

For selecting the visual features, we make use of a pretrained Resnet [2] model which is pretrained on ImageNet data. As it's pretrained on large scale ImageNet data, it is able to learn robust visual features and that is why we use it as a pretrained feature extractor. Resnet provides an alternative route for gradients to flow and does not suffer from vanishing gradient problem despite being a very deep convolutional neural network.

## Architecture

### Baseline Model

We tried with a baseline model which predicts the most popular product from the category in which the product belongs to from the scene. For example, if the query is for shoes which the user is wearing in the scene, we will recommend the most popular shoes. The performance of the model is shown in figure 6 below.

### Visual Compatibility Model

We use a visual compatibility based model, where we train the model to learn embeddings that learn to pair closely related anchor and product images together. This model is inspired by the one used in [4].

$$Compatibilty = \sigma(c - ||E_1X_1 - E_2X_2||^2) \quad (1)$$

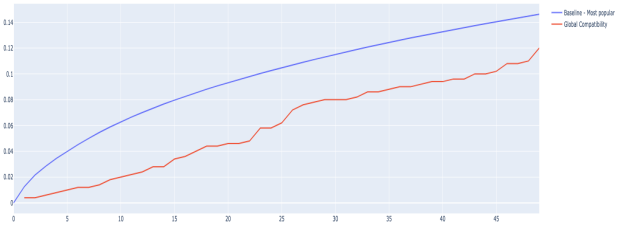


Figure 6: Top K accuracy of our baseline model as a function of K

| Technique | Global Compatibility | Visual Compatibility |
|---|---|---|
| | 51.2 | 58.12 |

Table 2: Binary Accuracy

Here visual compatibility refers to the notion of what products can substitute the given product appearing in the scene image. We use a pretrained ResNet [2] model to extract features. We take the output from the last layer of the network resulting in a 2048 dimensional feature vector. This is multipled with a embedding layer which is learnt during the training process. The task is portayed as a binary task, where the model tries to predict whether the given anchor and product pair is compatible or not.

### Global Compatibility and Triplet Loss

The architecture that we have used is described in fig. 7.

We make use of a pretrained Resnet [2] to get the output features, thereby using Resnet as a frozen feature extractor. Then we pass it to a 2 Layer Feedforward Network. The 2 Layer Feedforward Network (as shown in fig. 8)consists of a Linear Network which is passed through batch normalisation layer. The output of the batch normalisation layer is then passed through Relu layer. We have enabled dropout which will act as a regulariser and prevent overfitting. We then pass the output of Relu to another linear layer and we take the L2 norm of it. Figure 9 defines the process of in a lucid manner[1]. The ResNet model is initialized using the weights trained on the ImageNet task. We use the outputs from the last layer from the ResNet which is a 2048 dimensional feature vector.

Global Compatibility  To learn style embeddings, we measure the $l_2$ norm between the embeddings of the anchor(scene) image and the embeddings of the product images. Compatible images should have a lower norm value as their embeddings will be compatible.

$$d_{global}(s,p) = ||E_s - E_p||^2 \tag{2}$$

where $E$ denotes the embedding of the source and product images respectively. The loss function is defined as a triplet loss between the anchor image, positive product and negative product.

$$L = \Sigma[d_{(s,p^+)} - d_{(s,p^-)} + \alpha] \, \forall (s, p^+, p^-) \in T \tag{3}$$

## Implementation Details and Challenges

### Scalability and Memory Issues

We have used pretrained ResNet architecture as the feature extractor in our models. The weights of this ResNet architecture have been initialized with the ImageNet features. Owning to the size of the dataset used, the training of the models was done on Google Colab. We faced challenges with the training of the dataset as we were getting frequent out of memory errors. To avoid these errors, we purchased Google Colab Pro subscription in order to get better RAM(upto $25GB$) and Hard Disk space along with better GPUs and longer runtimes on terminals. Due to the large amount of data, we have trained our Global Compatibility model for 3 epochs.

### Triplet Loss

We extract the data using the script [2] but encountered Error 403: Forbidden code on 17 products. We used exception handling to handle such cases. We have used the Triplet loss function as described above. To compute this loss, we first implemented it on our own but this attempt turned out be unsuccessful as the loss function did not converge. We then implemented the loss function as given in the pytorch library [3].

## Literature Survey

We found a few research papers on visual recommendations for fashion data. We went through Kang et al. [3] which discusses about global and local compatibilities. Global compatibility is compatible to the equations 2 and 3. It also mentions about calculating local compatibility where we divide the scene into various parts and check the compatibility of each region with the product. Not all regions are equally relevant and the extent to which a particular region is relevant is determined using attention mechanism. It also mentions baseline models such as the popularity which recommends the most popular products from the same category as the product which is appearing in the scene. Another baseline which the paper mentions is using Siamese Networks. Pretrained Siamese Networks are used to learn image embeddings and the compatibility of the products is measured using l2 norm. The products having high compatibilities have a low l2 norm while products having low compatibilities will have a high l2 norm score.

We also went through McAuley et al. [4] which calculates compatibility between two products using shifted

[1] https://tinyurl.com/38r3ykrj

[2] https://github.com/kang205/STL-Dataset

[3] https://tinyurl.com/bddwapku

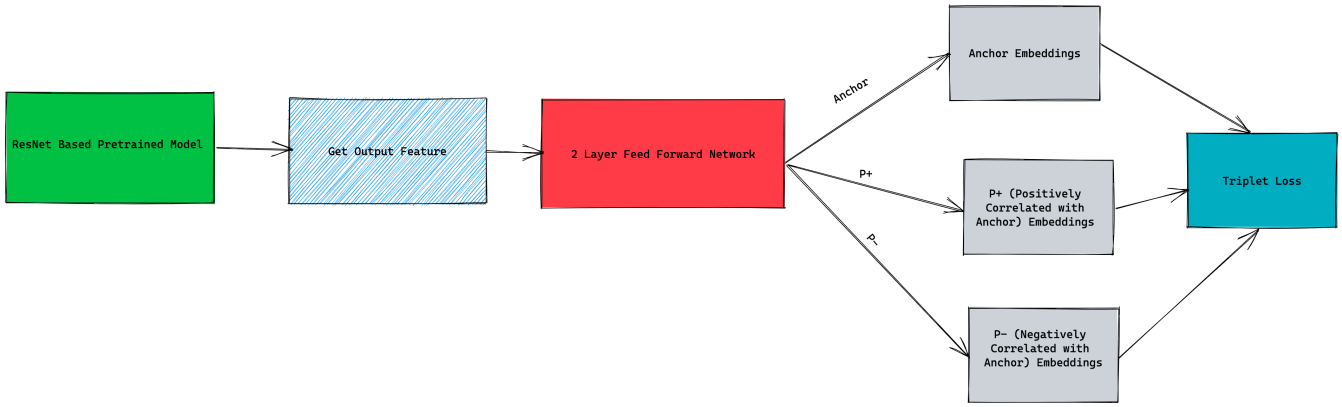


Figure 7: Architecture used. We use a Triplet Loss using an anchor(scene) image, positively correlated product image and a negatively correlated product image.

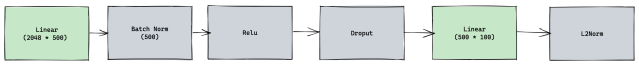


Figure 8: Architecture of the 2 Layer Feed Forward Network

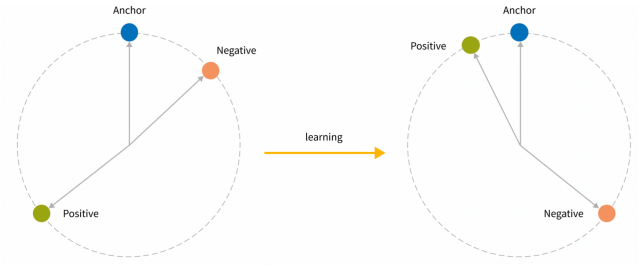


Figure 9: Triplet loss forces the positive image embeddings closer to the anchor image and the negative image embeddings to be farther away.

and inverted sigmoid transform of the distance between the two objects. The distance between the two objects is first calculated using the weighted nearest neighbours where the weights are learned so that more important features are given higher weightage. Weighted nearest neighbor is not able to capture subtle features needed to capture visually compatible products and mahalonobis distance is used to resolve that.

One of the state of the art models in visual recommendation is the one proposed in [5]. The authors have used an aggregated item representation using all the samples to create a robust representation of the images and then they use the triplet loss over these representations. This method achieves a Top-50 accuracy of 79.2 on the ExactStreet2Shop dataset.

## Results and Conclusions

Figure 6 mentions the top K accuracy performance of our global compatibility model and the baseline model which is based on popularity. Our Global Compatibility provides a maximum accuracy of 0.12 for K = 50 while the baseline model provides an accuracy of 0.146. Our Global Compatibility model performs worse than the baseline because we trained it on a limited dataset (20000 samples). We had 72,198 samples available but we could not make use of such a large dataset becausse we were getting frequent issues while loading the dataset as we were running it on Google Colab. Training time was also large and hence we trained our global compatibility for only 3 epochs. Also, during testing, we calculated the compatibilities for only top 200 most popular products from the catgeroy in which the product appearing in the scene belonged to. This is because we were getting frequent out of memory issues while trying to calculate similarities for each product present in the category. Table 2 provides the accuracies of our visual compatibily and global compatibility models where accuracy is calculated by calculating whether the given anchor and product images are compatible or not. Visual compatibility model achieves an accuracy of 58.12 and our global compatibility model achieves an accuracy of 51.2% This is because the global compatibility model is a large deep learning model which requires ample data to fully capture the semantics of the data which is difficult in our case due to the memory issues as mentioned above.

## References

[1] Hadi Kiapour, M.; Han, X.; Lazebnik, S.; Berg, A. C.; and Berg, T. L. 2015. Where to buy it: Matching street clothing photos in online shops. In Proceedings of the IEEE international conference on computer vision, 3343–3351.

[2] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.

[3] Kang, W.-C.; Kim, E.; Leskovec, J.; Rosenberg, C.; and McAuley, J. 2019. Complete the look: Scene-

based complementary product recommendation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10532–10541.

[4] McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, 43–52.

[5] Wieczorek, M.; Rychalska, B.; and Dąbrowski, J. 2021. On the unreasonable effectiveness of centroids in image retrieval. In International Conference on Neural Information Processing, 212–223. Springer.