Image Based Recommendation System

Nidutt Bhuptani
Khoury College of Computer Sciences
Northeastern University
Boston, United States
bhuptani.n@northeastern.edu

Anushka Patil
Khoury College of Computer Sciences
Northeastern University
Boston, United States
patil.anu@northeastern.edu

I. Introduction

With the advent of innovation and competition in the market, customers are favored with a heap of choices to choose from. Be that as it may, this untimely favoring can turn out to be a bane. 'Information overload' calls for sifting and organizing these alternatives for shoppers; hence—recommendation systems. The image of the item is the first thing that catches one's eye; in this way, it would be natural to deem pictorial representation as the gold standard for analyzing the 'content' of the item.

Online shopping has made the process of shopping convenient and less time consuming. Adobe has predicted that for the year 2022, online sales will be between $180 billion and $930 billion. [1] Recommendation systems first analyze the data that is available and use algorithms to suggest relevant items to the users. Recommendation systems can not only help improve user experience but can also be used to increase sales. A large number of online shopping search engines use key word matching in order to find a product that the customer would be interested in buying.

Currently, Amazon has implemented text based recommendation systems. Hence given a key phrase, the website fetches relevant products. This project is an application based project with the objective of building an Image based Recommendation system which will use image similarity of products to suggest products that the user is more likely to buy. We use latest State-of-the-Art CNN model (Resnet50) to achieve these results. To reduce the search time, we also implement different dimension reduction techniques- PCA, Kernel PCA, Isomap, Umap and t-SNE per say. We evaluate our engine using Precision@K.

II. Related Work

There exist two main approaches for product recommendation - collaborative filtering and content-based filtering. The former requires historic user-item data but the latter is normally good for a cold start problem where we recommend products based on item description. A newer deep learning based method is the neural collaborative filtering framework [2] which uses matrix factorization which is used in collaborative filtering. On the other end, other method [3] uses a hybrid approach, where a matrix factorization based predictor is mixed with Deep learning based method that will extract the embedding image features as well.

Deep learning has been used for quite some time now for content based product recommendation. A detailed survey of different deep learning techniques and their applications for recommendation was done by Sun [4]. He performed a detailed evaluation for a lot of Deep learning based techniques for this purpose such as RNN, CNN, autoencoders etc. CNN's have been used quite a lot for image recommendation in the last half decade or so. We know that successive levels of CNN represent image with increased level of abstractness. [5] won the first price for his work of using pretrained image feature extractor to generate recommendations. Shankar [6] proposed VisNet 1, a deep CNN architecture to learn embeddings to capture the notion of visual similarity, across several semantic granularities. This work was by one of the largest e-commerce website in India, Flipkart. They perform an analysis on 50M products and it's implementation is optimized to support 2K queries per second.

Siamese network [7] can also be used to retrieve similar objects. Here the network comprises of pair of CNN with shared weights where the model input is pair of images with the ground truth label as similar or not similar images. The main advantage of using Siamese network is that it is trained directly for the purpose we are trying to solve i.e to retrieve similar images. This is better than using models that are trained on other tasks such as object detection and using transfer learning for image similarity. However, the biggest flaw of using Siamese network is that since it is trained on data for binary classification - it does fail to capture Fine-grained similarity. Image similarity using a triplet-loss function has also been explored by [8]. This is a bit better than contrastive loss as it is easier to label relative levels of similarity than absolute level.

Fashion image retrieval has also been extensively studied in [9], [10]. They compare a number of approaches, however the best approach they found is using a 2 layered neural network trained on final layer of CNN to predict if two images match. They have also made their dataset publicly available.

Of late, there has also been a lot of industrial focus on image retrieval in the industry. Jing et al [11] presents a highly scalable and cost-effective commercial search system at Pinterest. However, its image embedding is a combination of Deep-CNN embeddings plus its color signature - which leads to a poor expressive power. There has also been some work done on Multimodal(image+text) [11] embedding to improve image search quality. Another novel approach for image based recommendation system is the use of clustering. A lot of vivid applications apply here - for example Liangliang Cao [12] uses clustering to suggest tourists with appropriate destinations based on images. Another great use of clustering was done by Quilong [13] where he combined clustering with SVD to perform clustering with collaborative filter recommendation. Clear examination of clustering for image recommendation does show that there isn't quite a lot of work done around it in the last decade. This is mainly because with the advent of deep learning and collaborative filtering, they have made clustering not a viable option anymore for image recommendation.

A lot of these works don't put a lot of focus on dimension reduction. One of the main aim of our work is to do a comparative analysis around the accuracy and inference time trade-off for various dimensionality reduction techniques. We use t-SNE, UMAP, Isomap, PCA and Kernel PCA on our image embedding vector.

III. Background information

A. Deep Convolution Neural Network

The core component of any ML tasks relating to images is a CNN model in most of the cases i.e CNN is a class of deep learning most commonly applied to visual datasets.The core operation behind it is a convolution, which is a mathematical operation applied on two functions, which produces a third function. A typical CNN architecture has several such convolution operation layers.

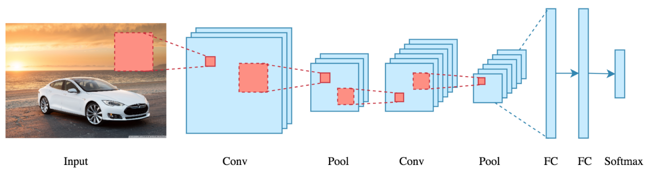

Fig. 1: A typical CNN architecture [16]

Figure 1 illustrates an example of CNN architecture. It may have many other different types of layers such as dropout, batch normalization, Average pooling etc, however the details of these are not much important for our approach. One thing to note is that the last 2 layers are where we have a vector of our input image. Most of the famous deep learning vision models have the last 2 layers as a vector of 2048 dimensions in the second last layer.

B. Resnet

Now that we have the basics set for CNN, the question comes to whether we should create our own CNN model or should we use transfer learning- it is clearly a design choice. We chose to go ahead with transfer learning primarily because we didn't have enough data to train our model from scratch and be able to outperform state of the art models. We went ahead with Resnet50 [15] which revolutionized the computer vision community. The reason being it solved the issue of training a deep neural network by introducing skip connections as seen in Figure 2 which allowed an alternate path to the next layers hence eliminating/reducing the vanishing gradient problem.

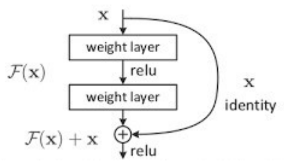
x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 2: Resnet50 skip connections [15]

C. Dimension Reduction

In most computer vision and NLP applications, we deal with high dimension data. One reason being high dimension data is often sparse because of curse of dimensionality and hence analyzing the data is often computationally expensive. Dimension reduction or dimensionality reduction is the transformation of data from a higher dimension space into a lower dimension space so that the lower-dimension representation retains some meaningful properties of the original data. We perform feature projection of data from the higher dimension space to space of fewer dimensions.
There are 2 main approaches to dimensionality reduction - one being the traditional matrix factorization which is good for preserving the global structure of data in lower dimensions while the other is building a neighbor graph which is good at preserving local structure of data in lower dimension.

*1) **Principal Component Analysis(PCA):*** One of the oldest dimension reduction [17] methods, it has 2 objective functions. One is to maximize the variance in the lower dimension. We compute the covariance matrix of the data and then compute the eigenvectors of the matrix. We then use the eigenvector corresponding to the highest eigenvalue to reconstruct a large fraction of the variance of the original data. In general, the first few eigenvectors contribute the most to the system's energy. Hence by a few eigenvectors, the original space with more dimensions has been reduced to a lower dimension with reduced number of data points.

*2) **Kernel PCA:*** Published [18] almost after a decade of PCA, it is an extension that uses Kernel trick that results in constructing non-linear mapping that maximizes the variance in the data. Many different kinds of kernels exist - linear kernel, polynomial kernel and gaussian kernel to name a few.

*3) **t-SNE:*** t-SNE [19] is one of the newer dimensionality reduction techniques which constructs a Neighbor graph for dimension reduction. It solves the issues of SNE technique by simplifying the cost function and using t-distribution in lower dimension to solve the crowding problem. One of the key parameter for this model is perplexity values which range from 5-50. Different values of perplexity change the cluster completely.

*4) **UMAP:*** Uniform Manifold Approximation and Projection (UMAP) [20] is a nonlinear dimensionality reduction technique which visually looks similar to t-SNE. However it is different because it assumes that there exists a dimension where the data is uniformly distributed. It also assumes that the underlying manifold of interest is locally connected. It constructs a graph of data in the higher dimension and then in the lower dimension. If the graphs are similar in both the dimensions, then we say the model has converged.

*5) **ISOMAP**:* This is a unique dimension reduction technique as it helps measure the distance along the manifold of the data. The input to our model is the distance between all pairs of data to each other. We use that to construct a neighborhood graph and output a weighted undirected graph. Post that we compute the shortest path between all pairs of points in our graph. Finally we compute r-dimension embedding by running PCA on the graph. ISOMAP enables us to transform our nonlinear data from its original input space to a new space, where PCA's linear dimensionality reduction is suitable.

D. Clustering

*1) **KMeans Clustering**:* One of the most popular and oldest clustering approaches, clusters object such that objects in the same cluster are more similar than the objects in other cluster.

*2) **Spectral Clustering**:* Spectral clustering is a technique that clusters non convex regions and allows for easily seperable boundaries. It does that by computing the eigenvalues of the similarity graph.

E. Cosine Similarity

Cosine similarity is one of the most widely used similarity measure. Given two vectors $\vec{v1}$ $\vec{v2}$, then its cosine similarity is given by:

$$\cos(\mathbf{v1}, \mathbf{v2}) = \frac{\mathbf{v1v2}}{\|\mathbf{v1}\|\|\mathbf{v2}\|} = \frac{\sum_{i=1}^{n} \mathbf{v1}_i \mathbf{v2}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{v1}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{v2}_i)^2}} \quad (1)$$

More the cosine similarity, better it is. Hence a similarity of 1 means the two vectors are very similar whereas a cosine similarity of 0 means the vectors are very different.

F. Evaluation

For evaluating the image retrieval we use two metrics - Precision@K. Precision at k is the proportion of recommended items in the top-k set that are relevant:

$$\textbf{Precision@K} = \frac{Number of recommended items @ k that are relevant}{Number of recommended items @ k}$$

Another famous evaluation metric is Recall@k. Recall at k is the proportion of relevant items found in the top-k recommendations.

$$\textbf{Recall@K} = \frac{Number of recommended items @ k that are relevant}{total of relevant items}$$

IV. Proposed approach

A. Feature extraction

After converting each image in our dataset into 240*240 dimension, we pass them to *Resnet50* architecture. We pass the data to all but the last 2 layers. Doing this allows us to obtain a 2048 dimension vector for each image which further acts as an encoder of our image.

B. Image retrieval using KNN

Once we have an image embedding for each image, we store it as a dataframe for future reference. This avoids having to compute embeddings of all the images in our dataset at inference. We tried storing pairwise similarity of all the images with all other images - this would allow O(n) complexity at inference. However, since the image dimension was 2048, we faced difficulty in doing so.
To recommend the top k images for a given query image from our dataset, we perform KNN on the embedding of all the images based on *cosine similarity*. The top k images with the least cosine similarity are returned to the user.

C. Dimension reduction

One thing we observed in the previous step is that each image had an inference time of about 9 seconds. This was primarily because each image was represented as a 2048 vector. Hence, we decided to reduce the dimension of our dataset. We started by performing a principal component analysis on our dataset. As evident from figure 5, 9.2% of the variance in our dataset is explained directly by the top 150 components. Also 100 components contributed to 86% of the variance due to which we decided to reduce our dataset to 100 features. We used 5 different dimension reduction techniques to understand the result on our recommendations:

1) Principal Component Analysis(PCA)
2) Kernel Principal Component Analysis(PCA)
3) t-SNE
4) UMAP
5) Isomap

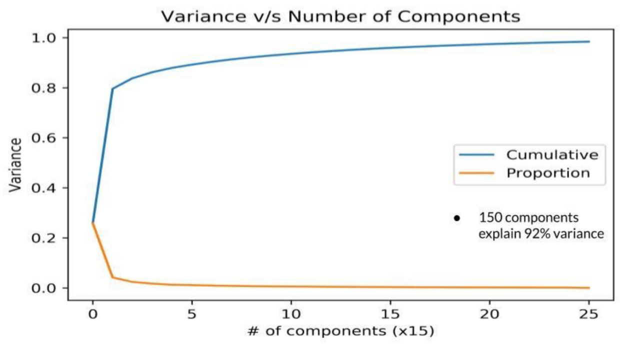

Fig. 3: Elbow point method for selecting the number of components to reduce our dataset to optimal dimension

D. Clustering

To validate the results of our model we also perform clustering on our dataset. We mainly performed two different types of clustering - kmeans and spectral clustering. Since the original image embedding is 2048 dimension, clustering on this high dimension data would lead to CPU exhaustion, we performed clustering on embeddings generated by Kernel PCA. To decide on the number of clusters, we do a check via elbow method for k-means clustering.

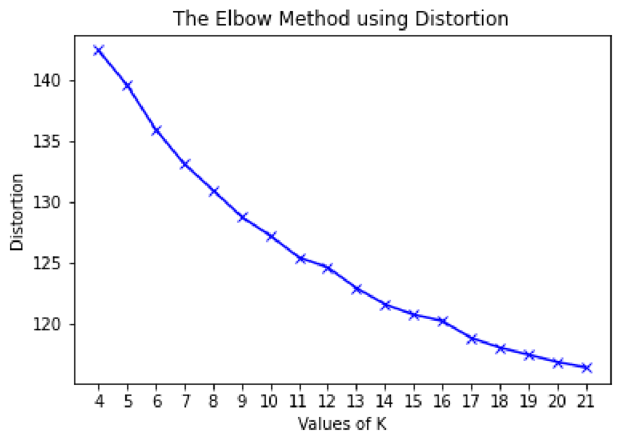

Fig. 4: Elbow point method to decide on the optimal K value

Since there is no evident break point here, we further investigate the effect of K on our evaluation metric Precision@6 which can be seen in figure 7. We do this for all 3 different masterCateogry to see if the model has more FP in one category than the other. We see below for number of cluster = 10, we have the highest average Precision@6(Note that the figure is for kmeans clustering)

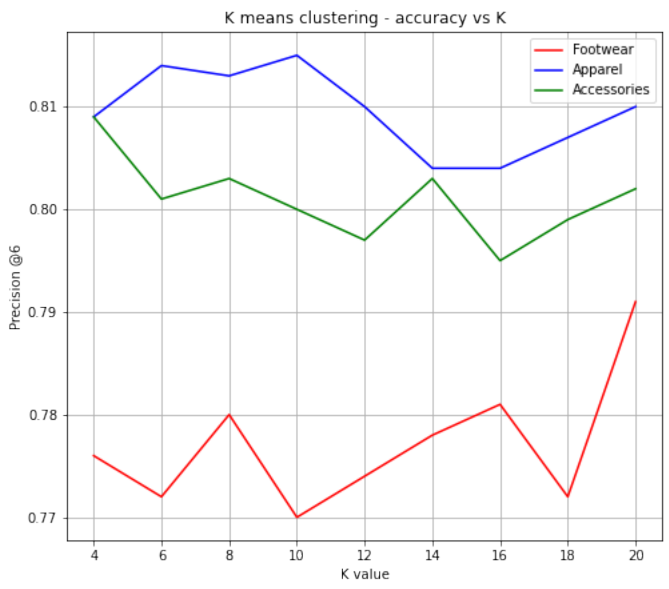

Fig. 5: Obtain ideal K value

We also tried spectral clustering with a set value of k=10 (Obtained in previous step). We didn't perform a proper exhaustive search for k like we did for K-means clustering because of lack of compute power.

E. Evaluation

For the purpose of evaluation we subset random images of all 3 master category - footwear, apparel and accessories.Each category has 100 images. Then we perform recommendation on all the 300 images. Each image returns 6 recommendations, and then we evaluate each returned image on a score of 0-1 with the following breakdown:

1) Gender : If the returned image is for the same gender as of ground truth.
2) Master Category : If the returned image has same master category(accessories, apparel and footwear) as of ground truth.
3) Sub Category : We check if the returned image is of same subcategory(topwear, bottomwear, socks etc).
4) Article Type : We check if the returned image is of same articleType(shirt, jeans, Watches etc).
5) Base Colour : We also check if both the images are of the exact same color.

So an image retrieved with all the 5 correct features is given full point(1) whereas the one which has 4 same features are given 0.8 points.

V. Experiments

A. Environment

While most of the implementation was done on Kaggle notebook(16GB CPU), the more compute heavy tasks such as dimension reduction was performed on AWS Sagemaker instance(ml.m5.12xlarge with 196GB RAM).

B. About Dataset

The dataset has high resolution images of products. It also has multiple label attributes that describe the products. Along with this, the dataset has descriptive text for describing product properties.

The dataset has 4 files :

1) Image Folder : Product Images in .jpg format.
2) Style Folder : Product Description in .json format.
3) images.csv : Publicly accessible URL for each image in our dataset.
4) styles.csv : Details about product's master category, subcategory, article type, color, season, year, usage and display name.

The dataset has 7 masterCategories, 49 subcategories, with 143 article types. Each product has a unique identification number associated with it, which can be used to map to all the products in styles.csv file. After this, the image can be found from the image folder with the same identification number. Metadata related to the image can also be fetched using this identification number.

Figure 3 illustrates the distribution of various **masterCategory** in the dataset whereas Fig 4 highlights the gender distribution.

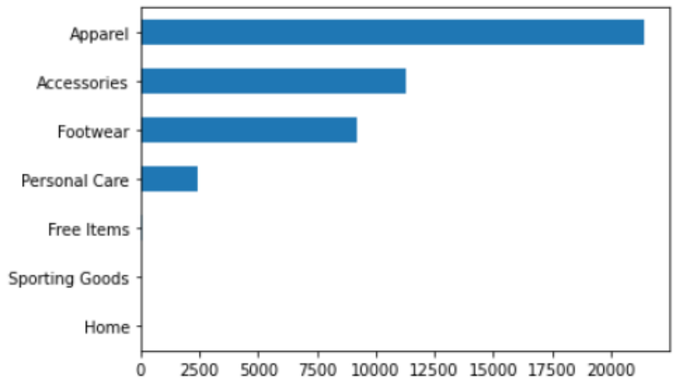

Fig. 6: Distribution of masterCategory in our dataset.

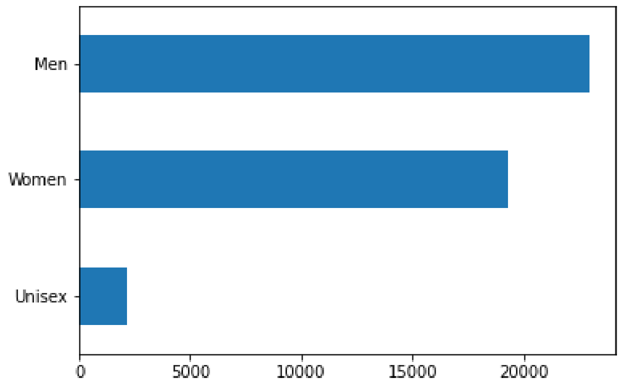

Fig. 7: Distribution of Gender in our dataset.

C. Data Extraction

Our first task was to do detailed analysis of the dataset to understand it. On the first glance, we observed only the top 3 masterCateogries - apparel, footwear and accessories have the most images. So we decided to use only these 3 categories - we discarded the other 4 - Personal care, free items, sports goods and house. Internally, between these 3 master categories, there are 19 different sub-categories. After we decided on the subset of the dataset, we observed each image was of very high dimension **2400*1800*3**. Hence we reshaped it **240*240*3** as it is the required input shape to **Resnet50** model.

D. Results

We performed our model evaluation using one metric - precision@6. We evaluated against this metric using several different embeddings - each corresponding to a different dimension reduction technique. A summary of the results can be seen below:

From the table it was quite evident that kernel PCA was best at encoding the images in a smaller dimension. Not surprising, PCA isn't that great here. UMAP and ISOMAP does a very good job as well. Another thing quite evident is that the clustering algorithms tend to have the least inference time.

Method	Footwear(P@6)	Apparel(P@6)	Accessories(P@6)	Inferenc
Resnet50 2048d	76.76	79.53	80.26	11s
PCA 100d	75.8	79.53	79.7	3.4s
KPCA 100d	77.4	81.23	81.94	3.4s
ISOMAP 100d	75	78.33	80.1	3.4s
UMAP 100d	77.23	79.34	80.23	3.4s
TSNE 100d	76.23	79.87	80.1	3.4s
Spectral Clustering	77.59	81.47	79.36	2.7s
kMeans Clustering	76.59	80.6	79.10	2.7s

TABLE I: Inference results

This is mainly because the search for the least cosine distance is restricted to only the items in the cluster. Since we have 10 clusters, so on an average the search space of image is also one-tenth.

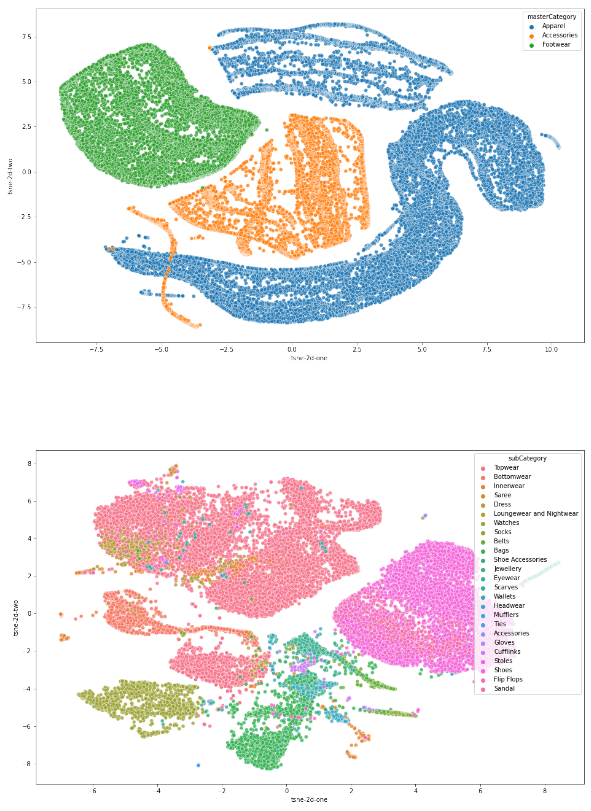

Fig. 8: Above 2d t-SNE visualization of masterCategory and below 2d Kernel PCA visualization of subCategory

We also decompose our embeddings to 2 dimension using the above mentioned dimension reduction techniques as seen in figure 8 and 9. We felt that it gives us a nice visualization and it helps us to better understand the data. We see that Kernel PCA does a very good job at maintaining the global structure of data whereas t-SNE preserves the local structure. We also performed 2D visualization using ISOMAP, UMAP and PCA but that led to poor results.

We also perform clustering to do image recommendation

- both kMeans and spectral clustering as seen in table 1. Among clustering algorithms, spectral clustering does a better job than kMeans clustering. However the train time of spectral clustering was much higher as compared to kMeans clustering. Hence we infer that in production kMeans clustering will be a much better option.

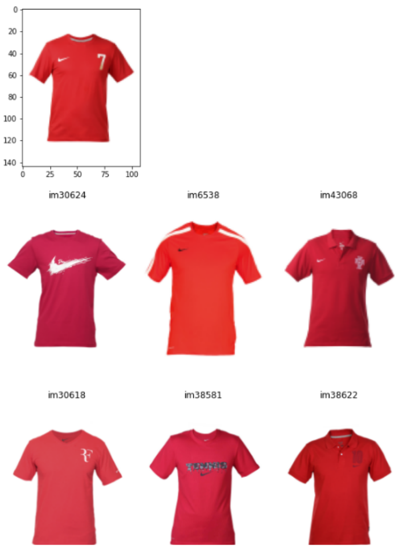

Fig. 9: Sample inference

Figure 9 shows a sample output of our kernelPCA based image recommendation. It does a very good job in 3 aspects - color, type of cloth and also the brand name. If you observe the reference image is a red tshirt of brand *Nike*. The returned images are also of brand *Nike*. This is really great as we were not just able to capture the global details of the image(color, category etc) but also the local details.

VI. Conclusion and Future Work

Content based recommendation systems are great for cold start problems i.e when we dont have any information about the users - about what kind of products they like , dislike etc. The various dimension reduction techniques discussed above reduce the inference time by more than 50% and in many cases still perform better. One reason for this might be the fact that our images consisted of human models a lot of the times. Reducing dimensions might eliminate their effect on an image encoding. We also observe that the system not only recommended images based on global pattenr in image but also based on local pattern as discussed in the previous section. Having said that, this is a very open-ended problem and a lot of work can further enhance the performance of the system.

1) Collaborative filtering : Last decade has shown that collaborative filtering leads to better results.We should collect some user data and implement it.
2) Classification : Approach the problem-set with supervised way as classification would lead to better results and accuracy. Products like Casual shoes and sports shoes will be better segregated or differentiated using Neural Networks.
3) Creating User interface : We can integrate our work and deploy it on a website for a real time testing of it.
4) Siamese Network : Siamese network are easy to train and generally do very good for content similarity tasks.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] He, X., Liao, L., Zhang, H., et al.: Neural collaborative filtering. In: WWW 2017, pp. 173–182 (2017)
[3] He, R., McAuley, J.: VBPR: visual Bayesian personalized ranking from implicit feedback. In: AAAI, pp. 144–150 (2016)
[4] Shuai Zhang, Lina Yao, Aixin Sun, Yi Tay: ”Deep Learning based Recommender System: A Survey and New Perspectives”, Arxiv, 2017
[5] Andreeva E, Ignatov DI, Grachev A, Savchenko AV,”Extraction of visual features for recommendation of products”, International conference on analysis of images, social networks and texts. Springer, Cham 2018
[6] Devashish Shankar, Sujay Narumanchi, H A Ananya, Pramod Kompalli, Krishnendu Chaudhury,”Deep Learning based Large Scale Visual Recommendation and Search for E-Commerce”,2017
[7] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In Proc. CVPR. 539–546
[8] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. 2015. Simultaneous feature learning and hash coding with deep neural networks. In Proc. CVPR. 3270–3278
[9] Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, Jiajing Xu, Jei Donahue, and Sarah Tavel. 2015. Visual Search at Pinterest. In Proc. KDD. 1889–1898.
[10] M. Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C. Berg, and Tamara L. Berg. 2015. Where to Buy It: Matching Street Clothing Photos in Online Shops. In Proc. ICCV
[11] Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, Jiajing Xu, Jei Donahue, and Sarah Tavel. 2015. Visual Search at Pinterest. In Proc. KDD. 1889–1898.
[12] Liangliang Cao; Jiebo Luo; Andrew Gallagher; Xin Jin; Jiawei Han; Thomas S. Huang, ”A world wide tourism recommendation system based on geotaggedweb photos”, IEEE, 2010
[13] Qilong Ba; Xiaoyong Li; Zhongying Bai, ”Clustering collaborative filtering recommendation system based on SVD algorithm”, IEEE, 2013
[14] Stevo. Bozinovski and Ante Fulgosi (1976). ”The influence of pattern similarity and transfer learning upon the training of a base perceptron B2.” (original in Croatian) Proceedings of Symposium Informatica 3-121-5, Bled.
[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun,”Deep Residual Learning for Image Recognition”, Arxiv,2015
[16] Shuai Zhang, Lina Yao, Aixin Sun, Yi Tay: ”Deep Learning based Recommender System: A Survey and New Perspectives”, Arxiv, 2017
[17] Pearson, K. (1901). ”On Lines and Planes of Closest Fit to Systems of Points in Space”. Philosophical Magazine. 2 (11): 559–572. doi:10.1080/14786440109462720.
[18] Schölkopf, Bernhard; Smola, Alex; Müller, Klaus-Robert (1998). ”Nonlinear Component Analysis as a Kernel Eigenvalue Problem”. Neural Computation.
[19] Schubert, Erich; Gertz, Michael (2017). Beecks, Christian; Borutta, Felix; Kröger, Peer; Seidl, Thomas (eds.). ”Intrinsic t-Stochastic Neighbor Embedding for Visualization and Outlier Detection”. Similarity Search and Applications. Lecture Notes in Computer Science. Cham: Springer International Publishing.
[20] Lawrence, Neil D (2012). ”A unifying probabilistic perspective for spectral dimensionality reduction: insights and new models”. Journal of Machine Learning Research.