Similar Product Search Using Images

Smit Anish Kiri, Nitin Kumar Mittal
[MSDS Spring 2020]

What is Image Based Product Search?

User wants to search for the following product:

Conventional method for product search includes providing textual description of product.

Textual Input: Brown Sunglasses

Suggested method for product search is directly allowing user to input picture of the product.

Pictorial Input:

Desired Output:

Recommending most similar products from the inventory.

Currently, the most widely used way to search for a product on a fashion e-commerce website is by providing its textual description. As a result, searching for similar fashion products becomes difficult. In order to obtain the desired outcome, the user is required to provide a textual description of the product that closely matches the textual context mapped to the product images stored in the databases of the e-commerce websites.

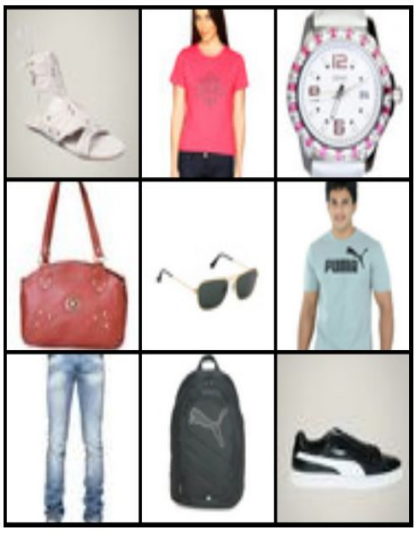
A much more difficult task is to find the exact product online. For example, if a user wants to purchase the same eyewear product shown above, conventional search methods will only allow users to provide a textual description which in our context can be brown sunglasses. Recommended products may or may not match the user search.

A much easier approach for the user to search similar products is letting the user use the fashion product image as an input. The desired output shown above contains the recommended products for the pictorial input of brown sunglasses.

In our project, we explored ways to identify and recommend similar products from existing product images in the database in real-time based on pictorial input from the user.

Available Dataset

1. Fashion Product Images.
    1.1. 44000 colored product images each of size 80 * 60.
2. Styles spreadsheet - image filenames and categorical description of products.
    2.1. 44 product categories like topwear, bottomwear, bags, watches, etc.
    2.2. Sampled 10800 images from 12-13 most frequent product categories for training.
    2.3. Assumed rough estimate of number of product categories.



Sample of Images in Dataset

The dataset used in the project can be divided into 2 types. The Dataset is obtained from Kaggle.

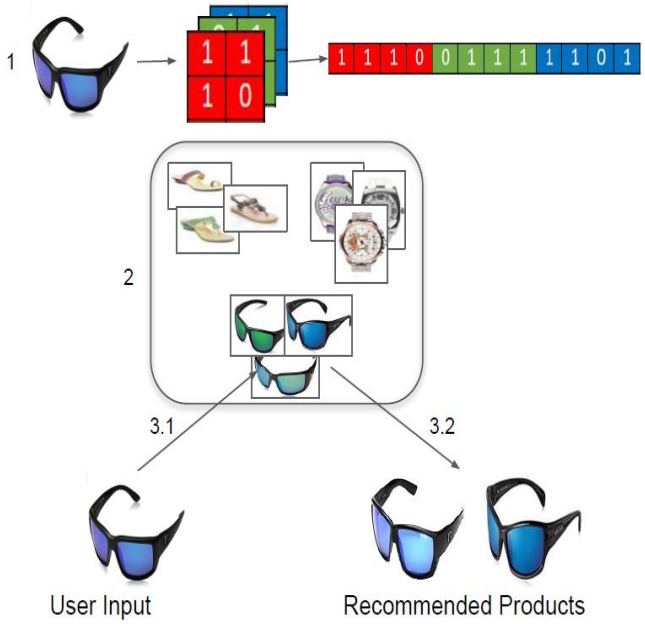
The first type of dataset consists of 44,000 fashion product images in JPEG format. Most images are of dimensions of 80 x 60 x 3 where 80 and 60 are the height and width of the images respectively and the third dimension represents the red, green and blue pixel values since images are coloured. Each image can be converted into integer arrays in order to perform numerical computations.

The second type of dataset is a CSV file which contains details about images such as categories (topwear, bottomwear, shoes, bags, etc) along with a short description of the images/products. We have made use of categories given in the CSV file to sample the dataset for training. Overall there are 44 categories along which the products are distributed. The number of products under the categories are distributed non-uniformly. Therefore, we sampled the categories for which the number of products was greater than 900 in the dataset. Overall we sampled 12 most frequent categories with an equal number of products sampled randomly. After sampling, our training dataset consisted of 10800 coloured images with 12 categories.

We assumed that the e-commerce vendors for whom we are building a similar product search engine have rough estimates of product categories that they have in their inventories.

Approach Followed to Build Image Based Product Search

1. Converted images into array of red, green and blue pixels.
2. Used clustering models to group product images.
3. 2-fold approach to recommend products:
    - 3.1. Using pre-trained cluster model to identify cluster for user's pictorial input.
    - 3.2. Recommending most similar products from identified cluster of images.



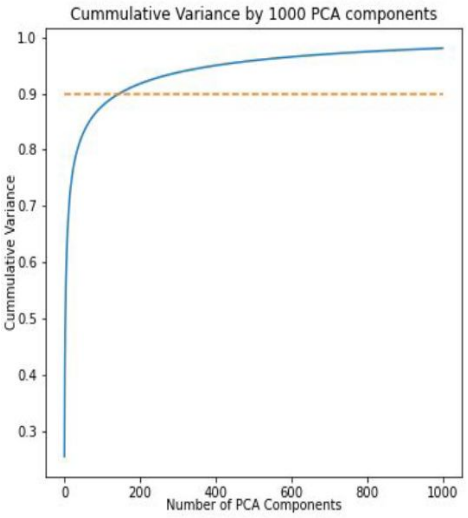
Here we described the approach that we followed to recommend the similar products to the user for their pictorial input.

The first step involved converting the image into a numerical computable format. Images were converted into numerical arrays consisting of red, green and blue pixel values.

In the second step, for the products that were present in the inventory/ training dataset, using their image pixel values, products were grouped together such that the products from the same category were clustered together. Clustering models like K-Means and Gaussian Mixture Model (GMM) were used to group similar products into clusters. In K-Means, similarities between products are calculated using Euclidean Distance between pixel values of their images. In GMM, Maximum Likelihood Estimation is used to calculate cluster distribution for every product again using their image pixel values.

In the third step, we followed a 2-fold approach to recommend similar products to the user's pictorial input. Firstly, we made use of a pre-trained cluster model to identify the most suitable cluster/ product category for a user's input product. Secondly, we searched and computed similarities between user's product and inventory products specifically present in the identified cluster. Here, clustering helped to reduce the search space. Without clustering, it is required to compute similarities between user input and all the products in the inventory. With clustering, the search space reduces to the identified cluster for the user input.

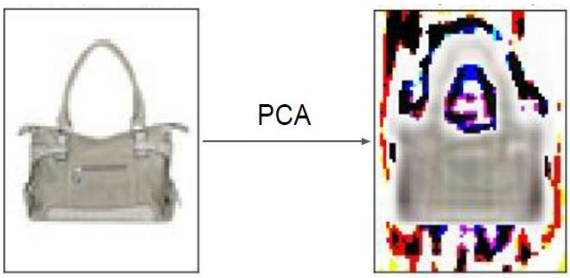
Image Dimensionality Reduction using PCA



Pareto Plot

145 principal components explained 90% of original image

Fig 1.1



Components in Original Image

80 * 60 * 3 = 14400

Reconstructed Image with 145 PCA components

Background destroyed but actual object preserved

Fig 1.2

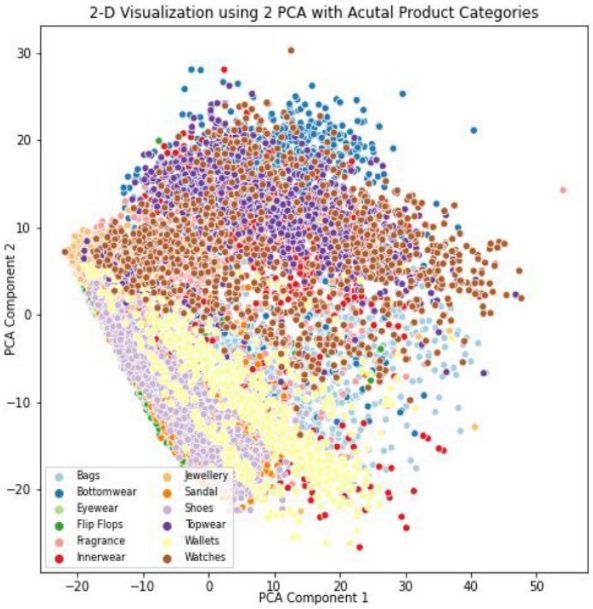
Working with image data is computationally expensive. As mentioned before, we performed clustering on product images that involve computing similarities between product images using their pixel values. Even for small-sized coloured product image of dimension 80 * 60 * 3, the total number of components after flattening the image array becomes 14400 which is huge.

In every iteration of K-Means and GMM with K clusters, for every image, similarity measures like Euclidean Distance and Likelihood Estimations are calculated for every cluster. Such operations become computationally expensive when dealing with large dimensional data. Converting higher dimensional data into lower dimensions without losing information helps to reduce the computational cost and yield real-time quality results.

We made use of Principal Component Analysis (PCA) as a dimension reduction technique. The Pareto plot [Fig 1.1] shown above describes the relationship between the number of PCA components and captured cumulative variance. We found that with just 145 largest principal components obtained from PCA, we were able to capture nearly 90% variance of the original data. Thus with PCA as dimension reduction technique we were able to drop image components from 14400 to 145. Reducing dimension by nearly 90% was one of the significant achievements to reduce the computational cost.

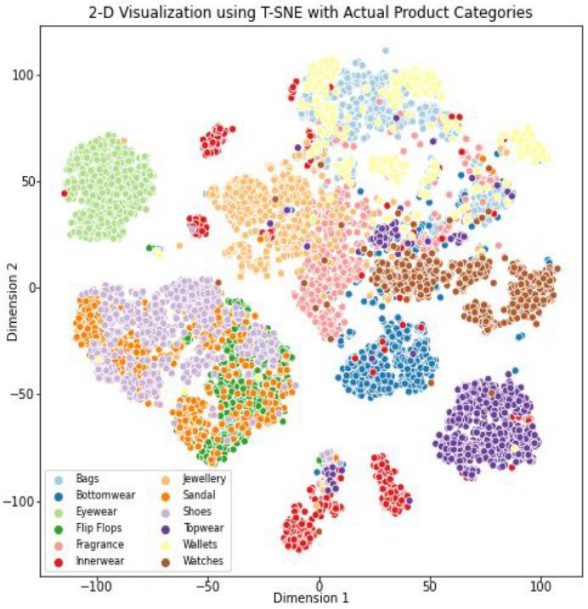
The reconstructed product image from 145 PCA components [Fig 1.2] shown above describes that we were able to preserve the product in the image, though the white background got distorted. The original sampled dataset was brought down from 14400 to 145 components using PCA.

2D Visualization of Product Images using PCA and T-SNE



Possible clusters of similar products cannot be visualized

Fig 2.1



Possible clusters of similar products can be visualized

Fig 2.2

To understand the possibility of clustering of products in inventory/ training dataset using their images, we made use of PCA and T-Stochastic Neighbor Embeddings (T-SNE).

2D visualization using PCA shown above [Fig 2.1] on the left was obtained by taking projections of product images on top/ largest 2 PCA components. From the plot, we found that 2D visualization using PCA was not able to explain possible underlying clusters of product images. No observable or distinguishable clusters were shown.

2D visualization using T-SNE shown above [Fig 2.2] on the right was able to explain possible underlying clusters of product images. We took advantage of available sampled categories from CSV to understand the distribution of product images and clusters. We saw that categories like eyewear, topwear, bottomwear and innerwear formed significantly distinguishable clusters. Products like sandals, flip flops and shoes that fall under the footwear category appeared to form a separate cluster though overlapping with each other. Overlapping between sandals, flip flops and shoes can be attributed to their common foot-like shape. Similarly, overlapping clusters were visible for categories like bags and wallets, which can be attributed to their common rectangle-like shape. Fragrance and jewellery and watches also appeared to form overlapped clusters that can be attributed to their common lustrous appearance.

Overall, 2D visualization from T-SNE helped us to understand the underlying clusters of products.

Clustering PCA Transformed Images using K-Means

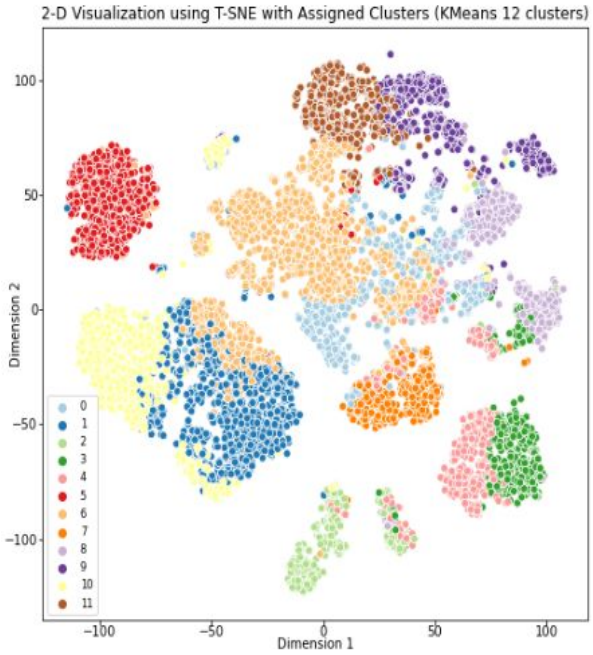


Fig 3.

1) Method

K-Means with 12 clusters.

2) Evaluation

Compared 2D T-SNE Plot with actual categories.

3) Results

Eyewear, innerwear, bottomwear and footwear categories formed distinct clusters.

4) Limitations

Overlapping - footwear subcategories (sandals, flip flops and shoes), topwear categories.

K-Means clustering was run on the sampled product image dataset after extracting 145 principal components using PCA. Before generating clusters, K-Means required the number of clusters K to which the data points were supposed to be assigned. The value of K was chosen to be 12 based on our knowledge of the sampled dataset. 2D visualization using T-SNE also gave a good proxy to estimate the optimum number of clusters to initiate clustering with.

After 300 iterations, cluster labels assigned by the K-Means algorithm for the sampled dataset were extracted and plotted on the 2D T-SNE plot. The plot shown above [Fig 3.] was then compared to the T-SNE plot [Fig 2.2] with the actual product categories to evaluate the performance of K-Means.

Upon comparison, we found that K-Means formed distinguishable clusters for categories - eyewear, innerwear and bottomwear. Significant overlapping was observed for all other product categories.

Clustering PCA Transformed Images using Gaussian Mixture Model (GMM)

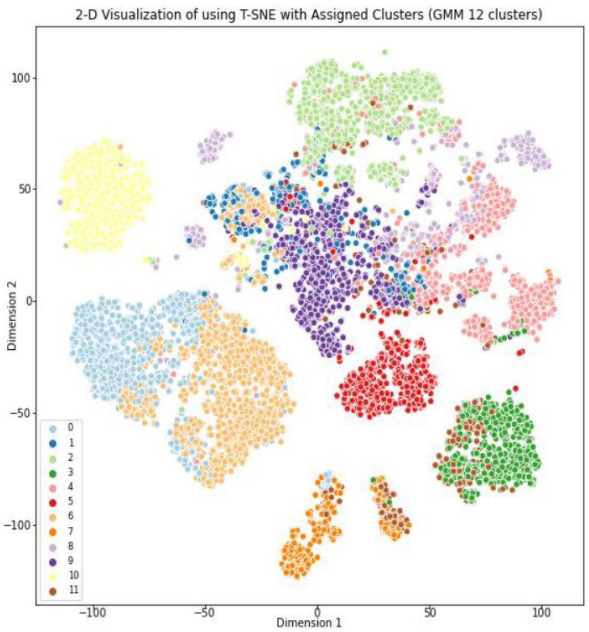


Fig 4.

1) Method

GMM with 12 components.

2) Evaluation

Compared 2D T-SNE Plot with actual categories.

3) Results

Eyewear, innerwear, bottomwear, footwear and topwear categories formed distinct clusters.

4) Limitations

Overlapping - footwear subcategories (sandals, flip flops and shoes)

After getting overlapped clusters from K-Means as per 2D visualization from T-SNE, Gaussian Mixture Model (GMM) was run on the sampled product image dataset after extracting 145 principal components using PCA. Similar to K-Means, GMM also required the number of components/ clusters K. The value of K was set to 12.

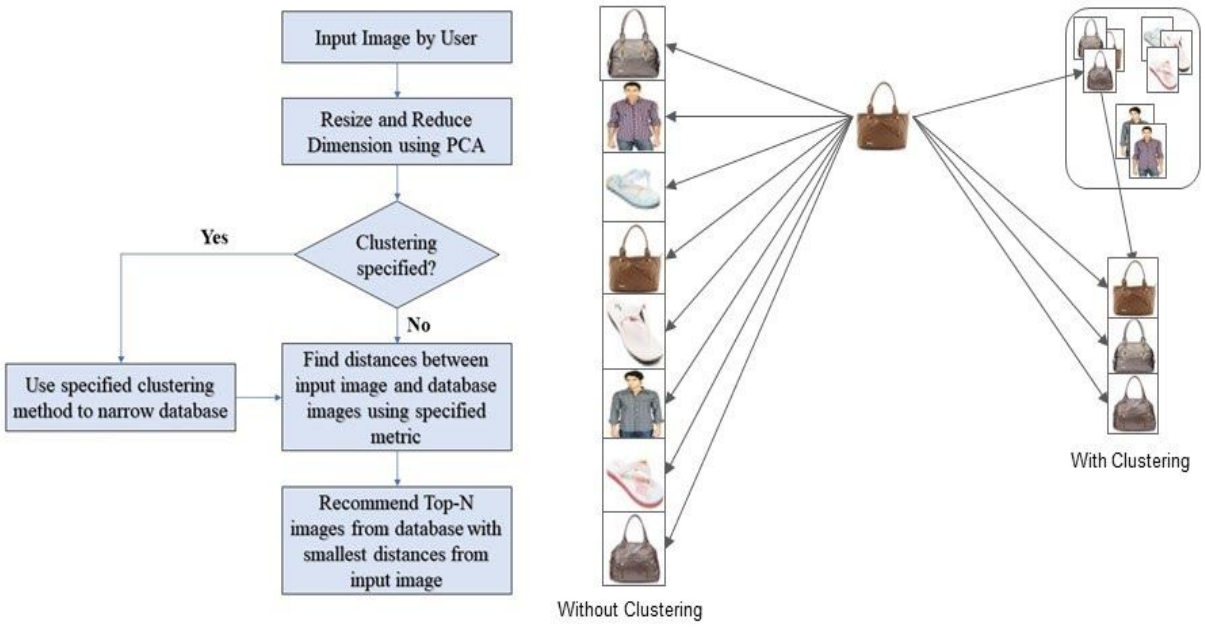
Unlike K-Means, GMM takes account of the variance of data points within the cluster and co-variance between different clusters. GMM takes advantage of likelihood maximization of data points to assign their cluster probabilities. For every data point (product image in our case), GMM provided soft assignments along with all clusters rather than hard assignments given by K-Means.

After 100 iterations, cluster labels (for every image assigning cluster label with maximum probability) assigned by the GMM algorithm for the sampled dataset were extracted and plotted on the 2D T-SNE plot. The plot shown above [Fig 4.] was then compared to the T-SNE plot [Fig 2.2] with the actual product categories to evaluate the performance of GMM.

Upon comparison, we found that GMM formed distinguishable clusters for categories - eyewear, topwear, innerwear and bottomwear. GMM was also able to assign separate clusters for products under the footwear categories like sandals, flip flops and shoes though observable overlapping within the footwear category was present.

Overall, we found that GMM performed better than K-Means to cluster similar products.

Recommending Images



The major part of our problem statement is to provide recommendations based on a fashion product image provided by a user. The recommendation process works as follows:

1. The user provides a coloured image of a fashion product as an input. The user also provides the number of recommendations / similar products N (default 5) as per their requirement.

2. The user input image is resized to 80*60 dimensions. The pre-trained PCA object is used to extract image projection on 145 principal components.

3. Clustering parameter is set optional from the backend. If a clustering method is specified, then it is used to predict a cluster label/ product category for the user input image and search space in the database is narrowed down to product images belonging to the predicted cluster label (product category). Specifying a clustering method helps to make the recommendation faster. If a clustering method is not specified, then the entire database is searched thus making the recommendation process slower.

4. Distance Metric is set to modifiable from the backend and used to compute similarities between the input product image and product images in the inventory.

5. Top-N images with the lowest distance / highest similarity with the user input image are recommended to the user in descending order of similarity.

Results and Optimal Metrics

Image

Colored, size 80x60

Dimensionality Reduction Technique

145 Principal Components using PCA

Clustering Method

Gaussian Mixture Model

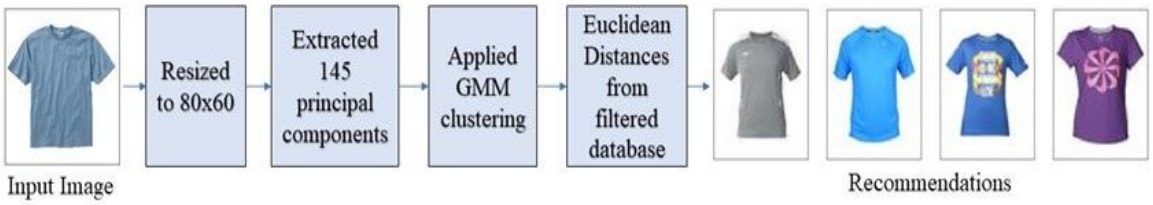
Distance Metric

Euclidean Distance

Evaluation

Randomly chose several product images from test dataset as inputs and observed the recommended outputs.

Obtained relevant recommendations.



After testing and comparing the recommendation results for product images with 1000 principal components and 145 principal components extracted upon PCA, the recommendation results came out similar, but the time and space complexity reduced significantly with 145 principal components.

To find the best combination of the clustering method and distance metric for the similar product recommendation process, nearly 30 test product images from different categories were used. Their recommendations were then observed and compared for various combinations of clustering methods and distance metrics. We found that the GMM method rarely classified the test image to a wrong cluster, thereby overall yielded the best recommendation results.

Manhattan and Euclidean distance metrics were tested and compared. It was found that Manhattan distance worked surprisingly well particularly for the topwear category but overall Euclidean distance gave the best relevant results.

Measuring the performance of a recommendation system is difficult. Using these metrics (GMM for clustering and euclidean distance as the distance metric) on randomly sampled images from the test set, the recommendation results were found to be relevant.

Challenges

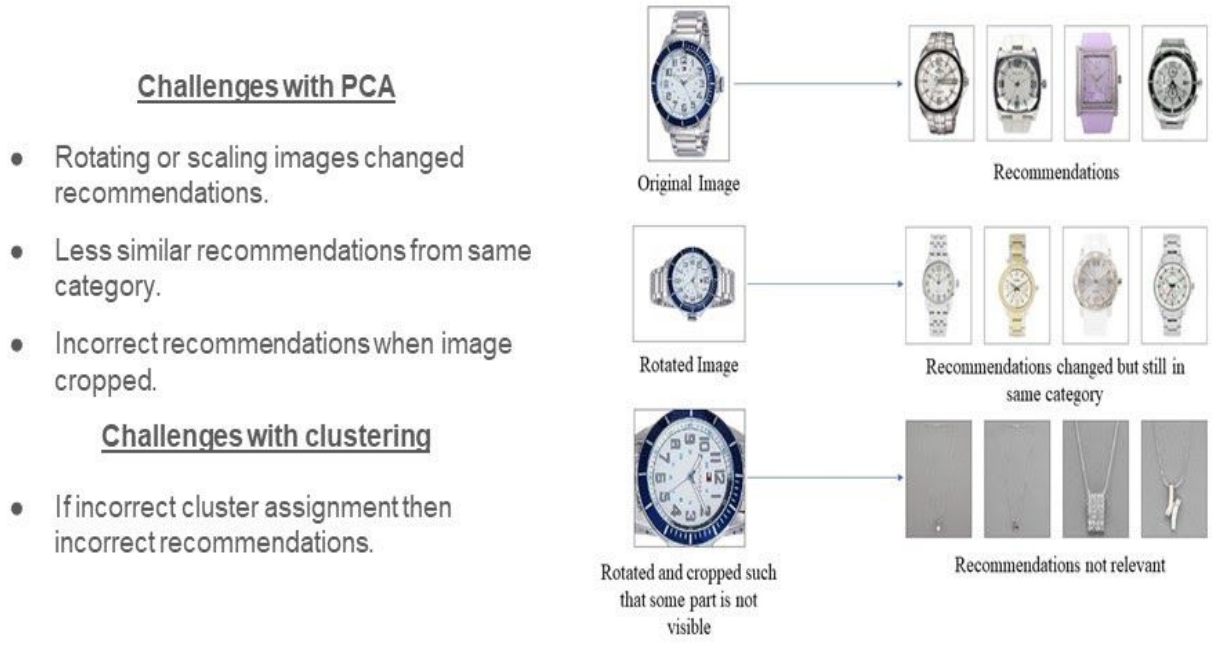


Fig 5.

One of the major challenges of extracting principal components for product images using PCA is their sensitivity to rotation and scaling of products in their images. It was found that even if the same product image is rotated or scaled by some amount, the recommendation results changed significantly.

Though even with the above stated sensitivity of principal components to rotation and scaling, the recommendations were still relevant, i.e the recommendations belonged to the same product category. However, if part of the product was cropped out of the input image, then the recommendations were completely different and irrelevant. For example, if the image of a wrist watch is cropped to remove the strap [Fig 5.] leaving just the dial, the recommendations changed to jewellery items. The same result was obtained even after turning clustering off in order to search the entire database to recommend similar products. Hence, it was concluded that the irrelevant recommendations were due to the PCA transformations and not because of clustering technique assigning input image to the wrong cluster.

Although the clustering method reduces the time complexity of the recommendation results, there is one major challenge with it. If the specified clustering technique assigns a wrong cluster to the input image, then the algorithm would essentially be searching for recommendations in the wrong set of images, thereby resulting in irrelevant results. Although, this was not observed with GMM clustering at most of the time. Though the search space can be expanded using the soft assignments capability of GMM clustering.

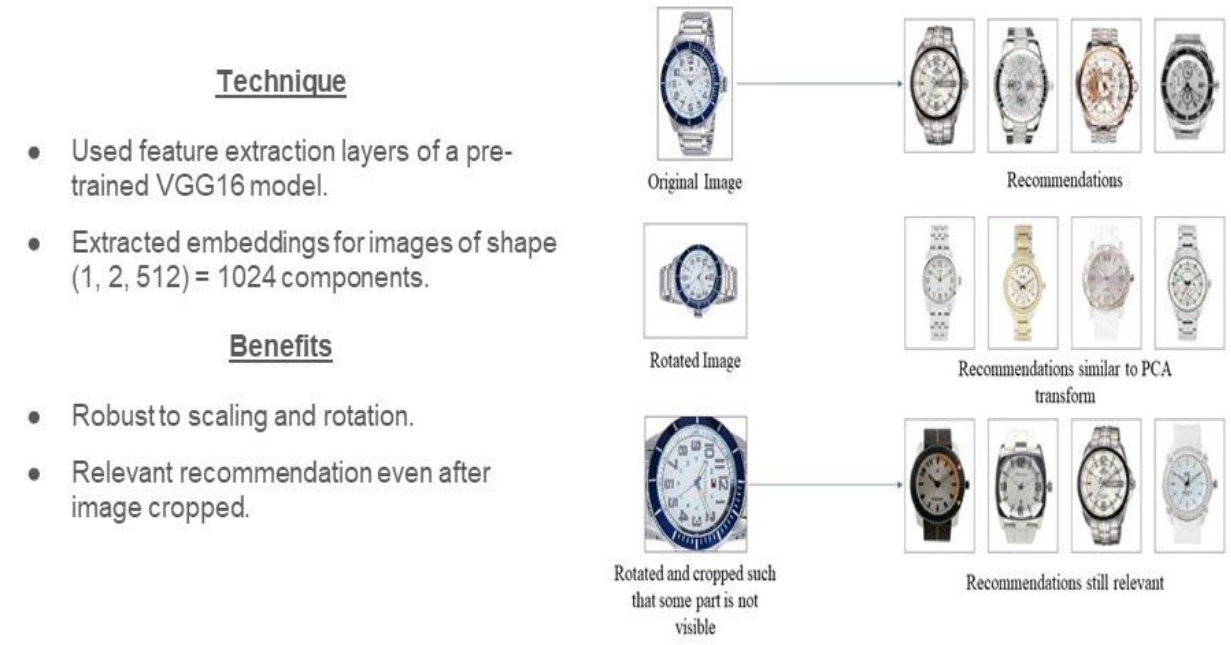


Fig 6.

To address the challenges of PCA transformations, we tried using embeddings for images. Using the feature extraction layers of a pre-trained VGG16 model, embeddings of shape (1, 2, 512) were obtained. The feature extraction layers of the VGG16 model basically extracts the features of an object present in an image, so if two images share some features, they will have similar embeddings.

It was found that using flattened embeddings (1024 components) instead of PCA transformations, the recommendations were much more robust to rotation and scaling of products in their images. Even if some part of the product if cropped out, relevant recommendations were obtained.

For example, if the image of a wrist watch is cropped to remove the strap [Fig 6.] leaving just the dial, the recommendations obtained were relevant. This can be explained using the fact that the embeddings are basically the features found in the images, so even if the image just contains the dial of a wristwatch, it still has plenty of features in common with other images of wristwatches.

In conclusion, the performance of an image-based search system can be improved using training dataset with versatile types of products images distributed uniformly along with available categories. Such dataset can help models to become robust to scaled, rotated and even cropped images of products and thereby recommending relevant output to the user.