Seminar Report

On

IMAGE BASED PRODUCT RECOMMENDATION SYSTEM FOR E-COMMERCE WEBSITES

BY

Bhavika Mauli Karale
Roll No. 178
Class TE-1

Under the guidance of
Prof. Pooja Dhule

‘ येथे बहुतांचे हित ’

DEPARTMENT OF COMPUTER ENGINEERING
Marathwada Mitramandals's College of Engineering Karve Nagar, Pune

Savitribai Phule Pune University

[2020-21]

DEPARTMENT OF COMPUTER ENGINEERING
Marathwada Mitra Mandal's
College of Engineering
Karve Nagar, Pune
Accredited with 'A' grade by NAAC

CERTIFICATE

This is to certify that **Bhavika Mauli Karale** from **Third Year Computer Engineering** has successfully completed her seminar work titled **"Image based Product Recommendation System"** at Marathwada Mitra Mandal's College of Engineering, Karve Nagar, Pune in the partial fulfillment of the Bachelor's Degree in Engineering.

Prof. Pooja Dhule **Internal Guide**	Head of the Department	Principal

Place:
Date:

Abstract

Over the last decade, India has witnessed an explosion in the e-commerce industry. There is increasing adoption of e-commerce in smaller towns and cities. As more and more business are getting connected through the Internet, a lot of similar products are available for sale, and hence there is a need for better search results. Most of the existing e-commerce websites are largely dependent on keyword matching and user history as their searching mechanism.

A convenient and reliable way of searching, is image based searching. It is an efficient and interactive way for querying related products, since product description displays a broad range of variation from supplier's side to receiver's side. The user will provide an image, and similar image based products will be presented to the user.

Machine learning model is used to classify the input image as one of the product categories. Then another neural network is used to calculate the similarity using methods like Jaccard similarity, cosine similarity, etc., which is used to select the closest product from dataset. Overall process is discussed to demonstrate the efficacy of image based search for product recommendation in practical applications.

Keywords

Random Forest, SVM, JPEG Compression, KNN, Cosine Similarity.

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude towards my esteemed guide Prof. Pooja Dhule for giving me this splendid opportunity to select and present this seminar and also providing facilities for successful completion

I thank Dr. H.K Khanuja, Head, Department of Computer Engineering, for opening the doors of the department towards the realization of the seminar, all the staff members for their indispensible support, priceless suggestions and for most valuable time lent to as and when required. With all the respect and gratitude, I would like to thank all the people who have helped me directly or indirectly.

Bhavika Mauli Karale
Class- TE-1
Roll Number- 178

Contents

List of Figures

List of Tables

1 TECHNICAL KEYWORDS

1.1 Domain Name

Artificial Intelligence

1.2 Technical Keywords

Recommendation System, CNN, Cosine Similarity, Deep Learning, Machine Learning, VGG Model, CNN Layers, Classification, Feature Extraction, Feature Vectors, Web Scraping, Data preprocessing, Wordcloud.

2 INTRODUCTION

The e-commerce world is fast evolving and is certainly growing around the world. Global retail sales growth will continue to rise and have increased their retail market share. For instance, according to eMarketer, e-commerce sales might achieve $6.39 trillion, taking up 21.8% of total retail sales. Even though all kinds of business had a tough time in 2020, national markets which are covered by eMarketer presented tremendous e-commerce growth. For example, Latin America had a huge growth (36.7%), even after considering a 3.4% drop in overall business. The e-commerce sector of India had a growth of around 35% in the last quarter of 2020. Argentina saw a growth of 79% in 2020, which is followed by Singapore, at 71.1%.

2.1 Domain Description

The enormous amount of gathered digital information has lead to the challenge of information overload for online customers, which hinders their timely access to interested products on the Internet. Thus increasing the need for recommendation systems. Today's e-commerce companies use their own recommendation systems, which are generally text-based and rely on data gathered from search terms, purchase history, product category, items in cart information and many more. This necessitates the buyer to give product descriptions, which differs largely from the supplier's end to receiver's end.

An alternate approach to this problem can be given through the rapidly developing technology of neural networks, which can help shift the search paradigm from textual description to visual entry. A product snapshot will provide a detailed set of information including its brand, usage, appearance, etc. The application of artificial intelligence for image recognition is less widely used and remains considerably unexplored. Hence, a smart recommendation system is studied which requires image of objects as input rather than a textual entry as input.

2.2 Problem Definition

To design a system to extract the meaningful product features from large dataset to increase the efficiency of image matching

In an image based search, an algorithm uses an image of the required object as input. A machine learning model then studies and classifies the object into the category it might belong to. Further, the features extracted from the last layer are fed to similarity calculation model to find the closest product from the dataset. Precisely, the functionalities studied here which are achieved in the recommendation system are:

1. Classification: The objective here is to correctly identify the product category from a random image.

2. Recommendation: The objective here is to find the most similar product from given features and category of product from the dataset.

2.3 Motivation

Search engines have always relied on text data to categorise and give appropriate search results. The scenario has however changed since the introduction of image search by Google Images. Instead of simply analyzing text, users can be provided with ways to search by content.

Since computer vision AI is evolving and machines have begun to see and understand more, visual search implementation is growing widely. With introduction of visual search, a change in user's search habits have been noticed. Thus e-commerce websites need to make use of it since visual search is a gateway to new possibilities in cases where the buyer lacks clarity about how to describe a certain product, but a visual reference.

Thus machine learning in the back-end studies the product image dataset learning through features like shapes, colors, brands, and returns visually similar results according to user's requirement.

3 LITERATURE SURVEY

3.1 Existing Methods

In the below mentioned papers, various techniques and methods are used. The desired result is achieved by using different CNN models like Inception- v3, JPEG feature vectors, k-nearest neighbors algorithm, Jacard similarity.

3.2 Literature Review

3.2.1 Image-Based Service Recommendation System: A JPEG-Coefficient RFs Approach [1]

In this paper, on the basis of user interaction, an idea to search the products efficiently in an e-commerce system using image based searching techniques is proposed. The paper consists of two important parts, Part 1 and Part 2. The proposed recommendation system learns the category/class/type of the product in Part 1. For Part 2, it retrieves similar products matched with input image.

The system uses Machine Learning(ML) to study the image/product features and generates a learned model. Further this model is used to categorize the queried product. Further in Part 2, from a particular class of products, the Euclidian distance based on JPEG feature vectors is used to get the 20 most similar product images. Top 10 most relevant products are then retrieved from the proposed Structural-Histogram approach, which uses the image features.

In the ML phase(Part 1) where the product category is learned, the authors have employed a Random Forests (RF) meta-classifier because of its excellent performance and generalization capability. The JPEG coefficients are used for feature extraction.

The paper uses Amazon dataset containing 20 categories of products for its proof of concepts. For enhancing performance of the proposed recommendation system, the RF model is further integrated into a Deep Learning setup.

3.2.2 Image-based Product Recommendation System with Convolutional Neural Networks [2]

The authors in this paper focus on two major problems which they want to solve. Firstly, determining the category of the given image; and secondly, recommend the most closely matching product according to the given image. This paper is majorly Convolutional Neural Networks based.

To solve the classification problem which is the given input image is to be classified into one of the 20 categories, the paper proposes construction of deep convolutional networks like AlexNet and VGG model which are compared with a baseline Support Vector Machine model.

For feature vectors of images, the last fully connected layer in the classification model is used to solve the recommendation problem. There is one feature vector mapped to any images in the dataset, which is fed as input to the recommendation system. The steps included in this are feature extraction, similarity calculation with comparison between cosine similarity method and Jaccard similarity method and output(recommendation).

The paper uses Amazon dataset of product image including data from between May 1996 to July 2014, which has 9.4 million products, out of which 3.5 million products are useful since others lack images.

3.2.3 Image Based Fashion Product Recommendation with Deep Learning [3]

In this paper, the fashion products are recommended to the customer by automatically extracting the available information of the user. A Deep Learning framework is created which helps recommend products/ images of similar style and taste. This approach is tested by using a publicly available fashion dataset. The framework is divided into two stages.

In the first stage, a Convolutional Neural Network (CNN) is trained for solving tasks related to image classification. This neural network classifier is used as an image feature extractor which also behaves as an input for recommendations. Here, Convolutional Neural Networks are trained separately for the prediction of the category of the product and texture type.

In the second stage, the k-nearest algorithm (k-NN) is used for ranking in feature space. After the first stage, the remaining Convolutional Neural Networks are integrated and used to extract the feature vector. The k-NN is then used to search the closest relevant item to the feature space.

The paper proposes a method that requires a single input to receive a list of similar style recommendations. The framework used helps in increasing the performance by accurately matching to the customer's style.

3.2.4 Image Based Search Engine Using Deep Learning [4]

The paper proposes an architecture of Deep Learning for CBIR systems. They have applied Convolutional Neural Networks for studying feature representation from the data containing images. A pre-trained CNN model, that is, Inception-v3 model, a GoogleNet deep architecture is applied on the dataset. The trained CNN is then used to classify objects according to their classes and perform an analysis to return the most relevantly similar image to that of an input image.

CBIR indexes pictures by extracting the visible features such a shape and colour. These indexed features are responsible for the retrieval of images. The information of pictures is separated by multi-dimensional vector features which in turn forms a feature database. An input image should be provided by the user to fetch a similar image. This input image is modified by the image retrieval model into a representative model of feature vectors. The image is retrieved by studying input image and the vectors of pictures in the database.

The semantic gap between the image pixels and the semantics perceived by humans is solved by Deep Learning. CNNs are used to handle image data which is a two-dimensional grid of pixels. The property of CNN to be able to learn shapes, colours and textures makes it suitable for applying it in the image-based search system. The paper depicts a way to model a reliable image retrieval system that will manage a database of images in a precise manner.

3.3 LITERATURE SURVEY

The Following table shows the literature survey by comparing techniques propose in various references:

Table 1: Literature survey

Sr No.	Title of paper	Year	Author	Advantages	Limitations	Concept
1	Image-Based Service Recommendation System: A JPEG-Coefficient RFs [1]	2020	Farhan Ullah, Bofeng Zhang, Rehan Ullah Khan	For product's class learning, the RF classifer is used. For feature extraction from images, the JPEG coefficients are used as image features.	For performance enhancements, the RF model needs to be integrated into the DL setup.	Proposed model uses a Random Forest classifier in the classification phase and JPEG coefficient vectors are used to calculate similarity scores by means of Euclidean distance and Cosine similarity in the recommendation phase to provide most closely matching images to the use input.

Sr No.	Title of paper	Year	Author	Advantages	Limitations	Concept
2	Image Based Fashion Product Recommendation with Deep Learning [3]	2018	Hessel Tuinhof, Clemens Pirker and Markus Haltmeier	A neural network classifier is used as a data-driven, visually-aware feature extractor. Initialization strategies using transfer learning from larger product databases are presented.	Accuracy based on a small dataset. Problem of cold start not addressed.	In the proposed model, the authors train a convolutional neural network (CNN) which is used as a problem-specific feature extractor, where the features serve as inputs for the ranking system. The k-nearest algorithm (k-NN) is used for ranking in feature space.

Sr No.	Title of paper	Year	Author	Advantages	Limitations	Concept
3	Image based product recommendation system with convolutional neural networks [2]	2017	Luyang Chen, Fan Yang, Hequing Yan	A CNN makes predictions by looking at an image and then checking to see if certain components are present in that image or not. If they are, then it classifies that image accordingly	They completely lose all their internal data about the pose and the orientation of the object and they route all the information to the same neurons that may not be able to deal with this kind of information.	Authors have discussed how deep convolutional networks like AlexNet and VGG models with a baseline SVM model can be used to solve the classification problem and Jaccard similarity is used to solve the recomme-ndation problem.

Sr No.	Title of paper	Year	Author	Advantages	Limitations	Concept
4	Image Based Search Engine Using Deep Learning. [4]	2017	Surbhi Jain and Joydip Dhar	Use of transfer learning. Extracting the last-but-one fully connected layer from the retraining of GoogleNet CNN model served as the feature vectors for each image, computing Euclidean distances between these feature vectors and that of our query image to return the closest matches in the dataset.	The paper did not consider the problem of sparse data. The approaches considered would have lot of sparse data and did not mention how they would deal with such redundant sparse matrix.	The paper proposes an architecture of Deep Learning for CBIR systems. A pretrained CNN model, that is, Inception-v3 model, a GoogleNet deep architecture is applied on the dataset for classification. Further, Euclidean Distance is used as similarity metric and closely matching products are retrieved.

4 MATHEMATICAL MODEL

The input given to the model is 1196 images of height and width of 224 by 224 pixels. The total number of features successfully extracted from all the 1196 images is 4096

In [29]:
```
# compute cosine similarities between images

cosSimilarities = cosine_similarity(imgs_features)

# store the results into a pandas dataframe

cos_similarities_df = pd.DataFrame(cosSimilarities, columns=files, index=files)
cos_similarities_df
```

Out[29]:

	C:\Users\Admin\Desktop\seminar\images\\images0.jpeg	C:\Users\Admin\Desktop\seminar\images\\images1.jp
C:\Users\Admin\Desktop\seminar\images\\images0.jpeg	1.000000	0.8282
C:\Users\Admin\Desktop\seminar\images\\images1.jpeg	0.828224	1.0000
C:\Users\Admin\Desktop\seminar\images\\images10.jpeg	0.784712	0.8984
C:\Users\Admin\Desktop\seminar\images\\images100.jpeg	0.321055	0.3229
C:\Users\Admin\Desktop\seminar\images\\images1000.jpeg	0.220629	0.2024
...	...	
C:\Users\Admin\Desktop\seminar\images\\images995.jpeg	0.312015	0.2368
C:\Users\Admin\Desktop\seminar\images\\images996.jpeg	0.284390	0.2567
C:\Users\Admin\Desktop\seminar\images\\images997.jpeg	0.284390	0.2567
C:\Users\Admin\Desktop\seminar\images\\images998.jpeg	0.284875	0.2514
C:\Users\Admin\Desktop\seminar\images\\images999.jpeg	0.284875	0.2514

Figure 1: Cosine Similarity

Cosine Similarity

$$Cos\_Sim = \frac{A \cdot B}{||A|| \ \ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

A_i, B_i : input image vector components

The Cosine Similarity is also used as the similarity score where the output is scaled between 0 and 1.

Output

Results are obtained in the range 0 to 1, where 0 indicates 0% similarity and 1 indicates 100% similarity.

5 PROPOSED SYSTEM ARCHITECTURE

5.1 System Architecture

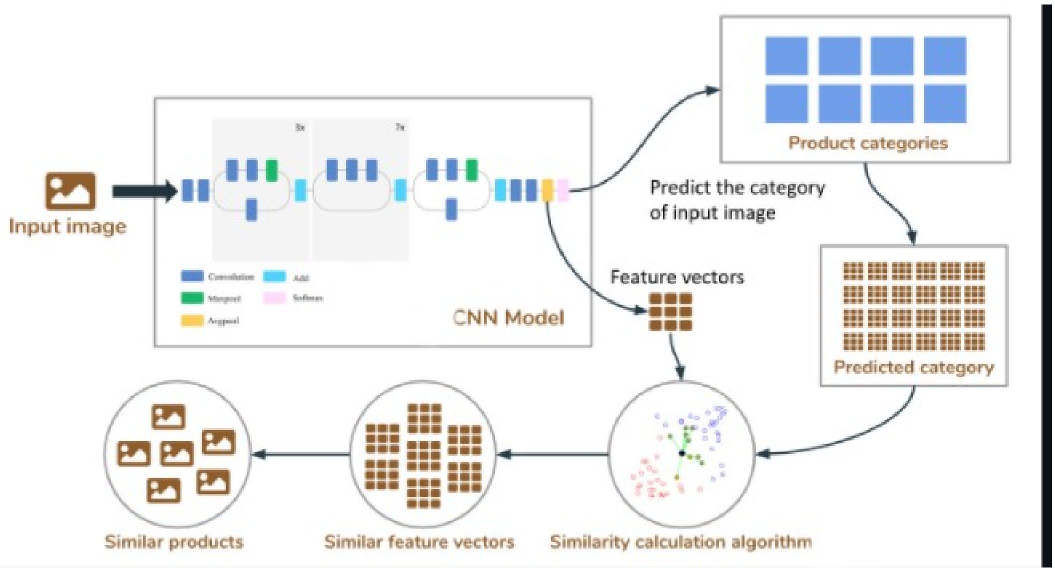


Figure 2: System Architecture

After performing data preprocessing on the data scraped from amazon.com using Parsehub, the images are given as input to the system. A CNN model is trained based on these input images. Further, features of the images are extracted using which the cosine similarity is calculated. Based on this similarity, the images are compared with the input image and the most relevant images are given as an output.

5.2 UML Diagrams

5.2.1 Class Diagram

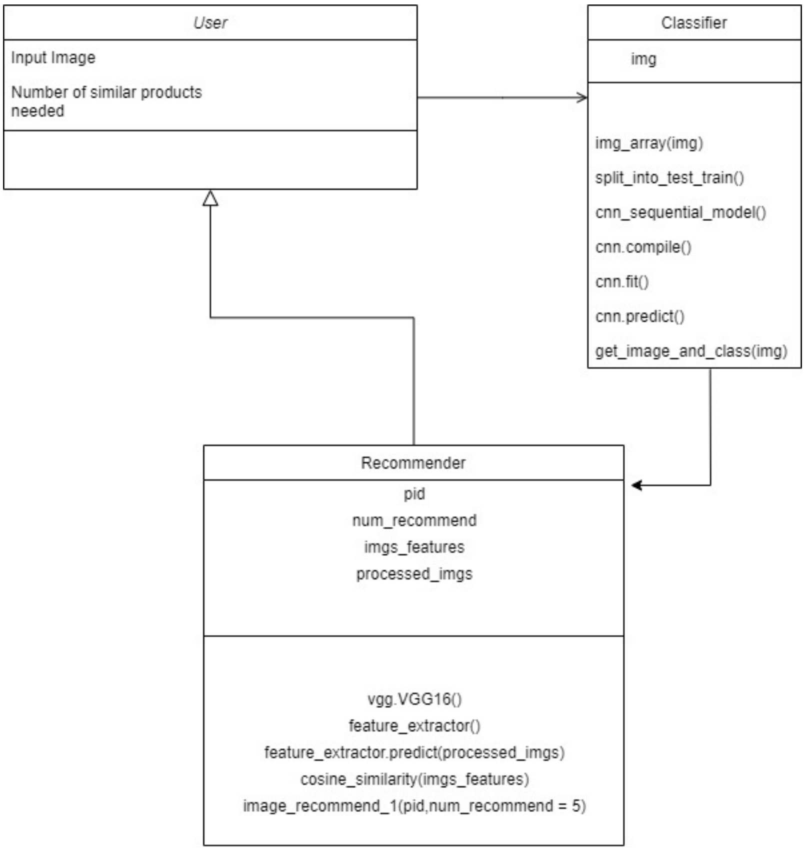


Figure 3: Class Diagram

Here, the cycle is divided into three parts.
In the first part, the user gives the input image and the number of similar images needed.
In the second part, based on the user's request, it goes through the classifier.
The third part extracts the features and checks for the similarity score and provides the user a valid output.

5.2.2 Activity Diagram

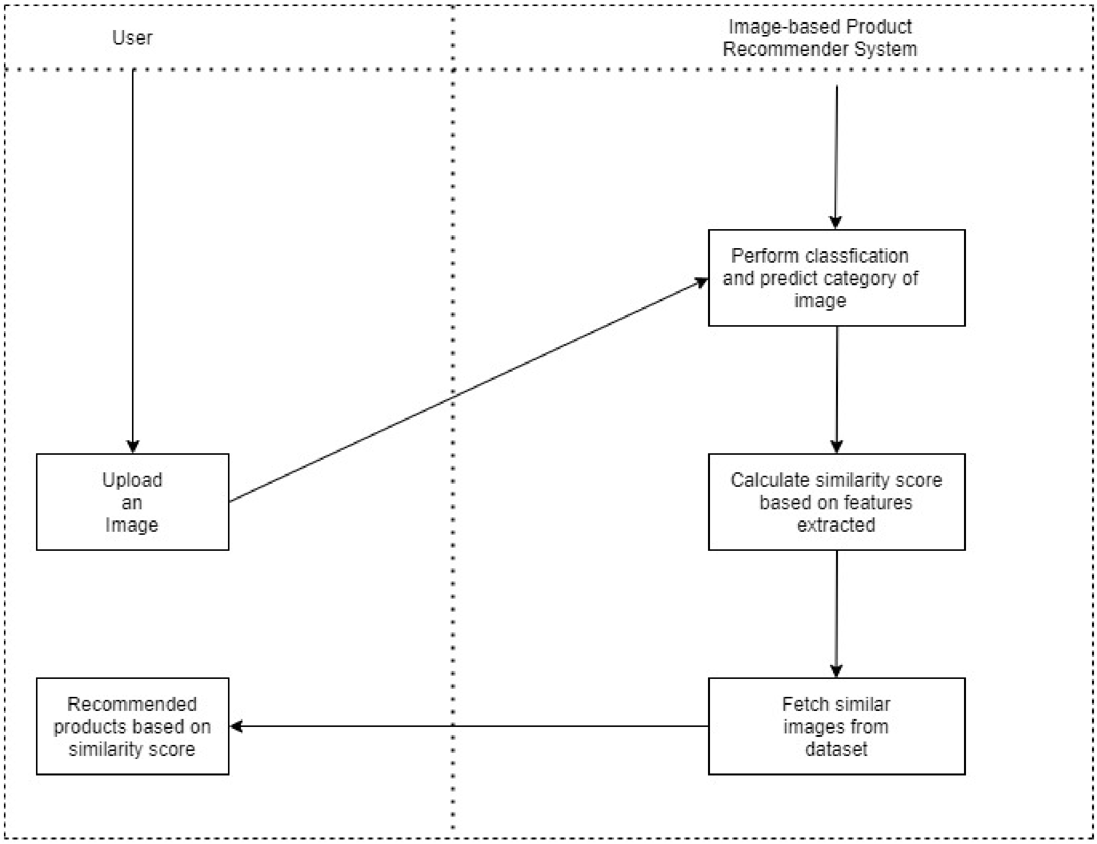


Figure 4: Activity Diagram

Here, the user uploads an image which is passed to the recommender system where the classification and prediction of the image based on its category and features is performed, features are extracted. Based on this classification and extraction, a similarity score is calculated which helps in fetching the similar images from the data which are provided to the user as an output

5.2.3 Use Case Diagram

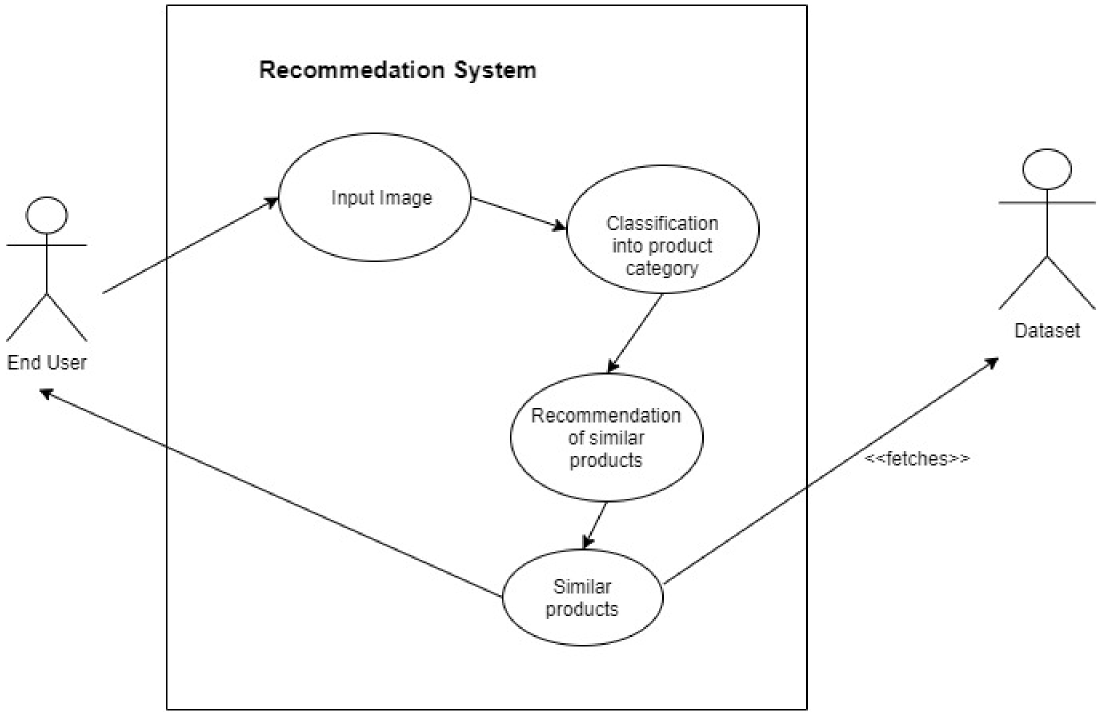


Figure 5: Use Case Diagram

The user provides with an input image which further gets classified into product category and its feature vectors are extracted based on which similarity score is calculated. According to this similarity score, similar images are fetched from the dataset and are provided to the user as an output

5.2.4 Deployment Diagram

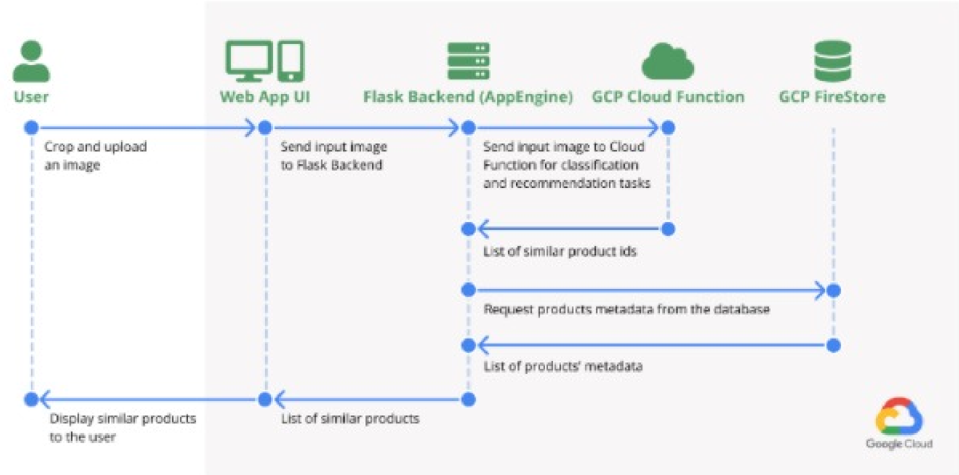


Figure 6: Deployment Diagram

The above diagram explains how the deployment of the final system can be performed by using Google Cloud Platform. VGG16 model is used for feature extraction which is of about 1 gb which makes it necessary for the product to be deployed on Google Cloud.

5.2.5 Component Diagram

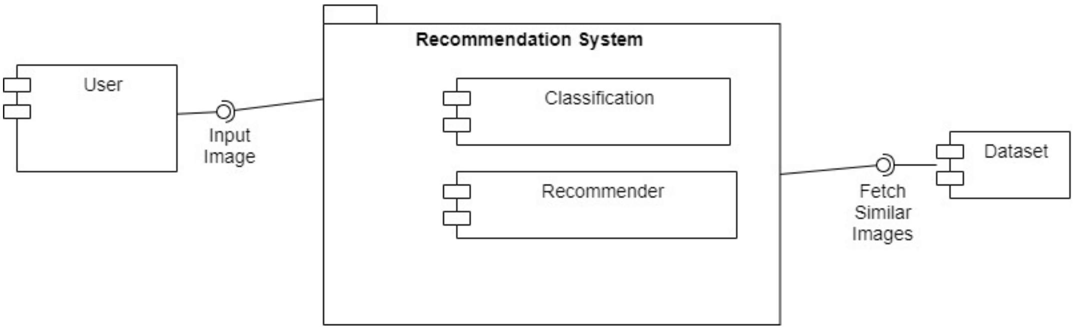


Figure 7: Component Diagram

The component diagram depicts the components of Image based recommender system. The recommender system has two modules, classifier and recommender. The classifier module is used to predict the category of the product from the input image and the recommender module further extracts features, calculates similarity and fetches the images of relevant images from the dataset.

5.3 ALGORITHM

The recommended system[1] uses a two phase, Phase 1 and Phase 2, algorithm to perform an image-based search to find products closest to the one selected by the user.

5.4 Phase 1: Algorithm for Image Classification:

1.Create a function to download the image links from the dataset.
2.Use the above function here to store all the images in the dataset into array.
3.Convert the response variable into numbers.
4.Split the data into training and testing sets.
5.Normalize pixel values to be between 0 and 1.
6.Set the seed and add the convolutional layers.
7.Add the flatten and last dropout layers.
8.Compile the model.
9.Fit the model.
10.Perform prediction on the test image.
11.Display confusion matrix for results.
12.Convert the test labels into a list.
13.Create a function which picks random images and identifies the class to which the image belongs.

5.5 Phase 2: Algorithm for Image Recommendation:

1.Load the model.
2.Remove the last layers in order to get features instead of predictions.
3.Print the layers of the CNN.
4.Load the images that were downloaded.
5.Use regex to extract only the pids from file names.
6.Add the pid and the image URL to a dataframe.
7.Load all the images and prepare them for feeding into the CNN.
8.Set the image size to 224*224.
9.Convert the images to array.
10.Extract the images features.
11.Compute cosine similarities between images.
12.Store the results into a pandas dataframe.
13.Create a function to retrieve the most similar products for the given one.

6 METHODOLOGY

6.1 Workflow

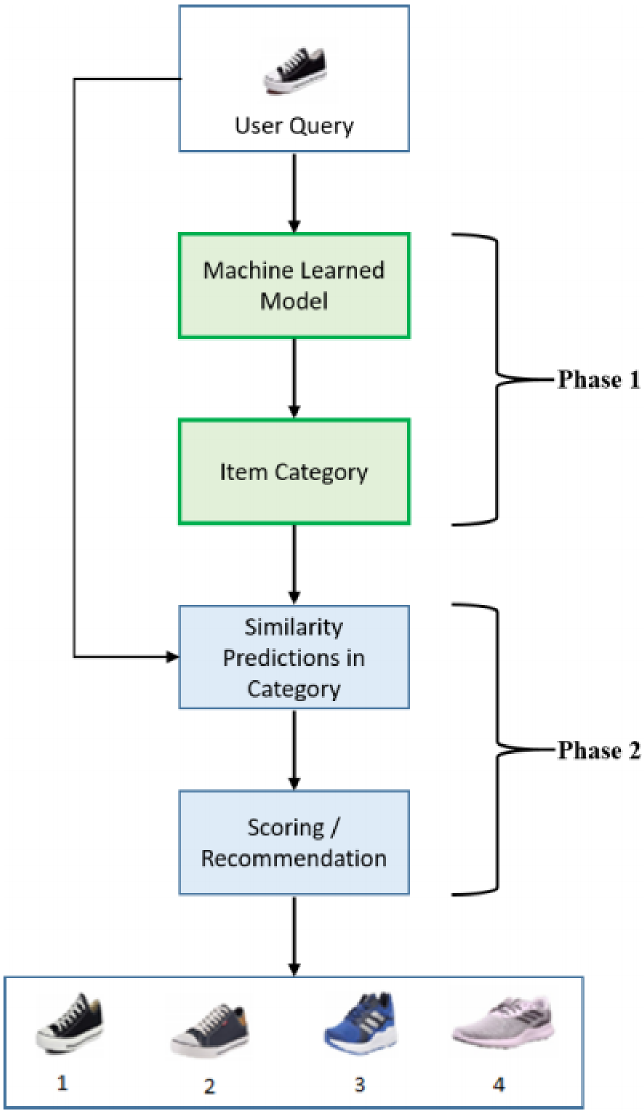


Figure 8: Workflow

The generic flow of the proposed recommendation approach. Green represents "Phase 1" which learns the category of the query image. The blue color represents the "Phase 2" which retrieves similar images from a particular category of images.

Phase 1: Category learning (Green color)
Phase 2: Recommendation (Blue color)

7 IMPLEMENTATION

The data was scraped from Amazon.com. A tool called parse hub was used for scraping the content. The 5 categories decided are - hats, shoes, shirts, pants and watches. The total number of results I got were 1960 rows.The 5 CSV files (per category) were merged and an output label column was created identifying the output classes. The scraped data has many missing values and discrepancies. An initial cleaning in excel was performed in excel.
The information includes the below: 1. Product title
2. Price
3. Number of reviews
4. Rating on a scale of 5
5. Image of the product

The further cleaning process and visualization was done using pandas library and matplotlib.pyplot respectively. An additional column called "PID" (Product ID) is added as a proxy for Product's ASIN number in Amazon

In [45]:
```
df=df.reset_index()
df=df.rename(columns={"index":"PID"})
df.head(5)
```

Out[45]:

	PID	Unnamed: 0	product_name	product_price	product_reviews	product_image	output
0	0	0	Men's Rugged Professional Cap	16.99	6,387	https://m.media-amazon.com/images/I/91tgXrB27K...	hat
1	1	1	Men's Athletic Baseball Fitted Cap	11.99	18,325	https://m.media-amazon.com/images/I/61pSuZP7SV...	hat
2	2	2	Men's Superlite Relaxed Adjustable Performance...	24.00	6,878	https://m.media-amazon.com/images/I/71qx1PZy3a...	hat
3	3	3	Unisex-Adult PFG Mesh Fish Flag Ball Cap	24.04	5,086	https://m.media-amazon.com/images/I/619C+F3rkp...	hat
4	4	4	Men's Ameritage Dad Adjustable Cap	22.00	6,541	https://m.media-amazon.com/images/I/818smS7zne...	hat

Figure 9: PID

The WordCloud Python library to create word clouds of the words in the title of each of the apparel type.

```
In [26]: for i in range(len(list_masks)):
             wordcloud = WordCloud(background_color="white",  mask=list_masks[i]).generate(product_join(list_name[i]))
             plt.figure(figsize=[5,5])
             plt.imshow(wordcloud, interpolation="bilinear")
             plt.axis("off")
             plt.show()
```

Figure 10: WordCloud

The objective of classification is to identify the correct category of product from a random image i.e. if its a watch, shoe, shirt, pant or a hat. CNN model is trained using a small training size to avoid kernel crash. To overcome the small training size(underfitting), I have increased the number of epochs during the compilation of the model.

```
Epoch 1/10
12/12 [==============================] - 95s 7s/step - loss: 0.8234 - accuracy: 0.7670 - val_loss: 0.8688 - val_accuracy: 0.833
3
Epoch 2/10
12/12 [==============================] - 94s 8s/step - loss: 0.5749 - accuracy: 0.8272 - val_loss: 0.7449 - val_accuracy: 0.833
3
Epoch 3/10
12/12 [==============================] - 122s 10s/step - loss: 0.4091 - accuracy: 0.8613 - val_loss: 0.5217 - val_accuracy: 0.8
333
Epoch 4/10
12/12 [==============================] - 114s 10s/step - loss: 0.4065 - accuracy: 0.8613 - val_loss: 0.4305 - val_accuracy: 0.8
750
Epoch 5/10
12/12 [==============================] - 115s 10s/step - loss: 0.3153 - accuracy: 0.8874 - val_loss: 0.3976 - val_accuracy: 0.8
958
Epoch 6/10
12/12 [==============================] - 109s 9s/step - loss: 0.2675 - accuracy: 0.8874 - val_loss: 0.4029 - val_accuracy: 0.89
58
Epoch 7/10
12/12 [==============================] - 114s 10s/step - loss: 0.2288 - accuracy: 0.9162 - val_loss: 0.3767 - val_accuracy: 0.8
854
Epoch 8/10
12/12 [==============================] - 114s 9s/step - loss: 0.2049 - accuracy: 0.9241 - val_loss: 0.2835 - val_accuracy: 0.92
71
Epoch 9/10
12/12 [==============================] - 114s 10s/step - loss: 0.1668 - accuracy: 0.9267 - val_loss: 0.2776 - val_accuracy: 0.9
167
Epoch 10/10
12/12 [==============================] - 114s 10s/step - loss: 0.1384 - accuracy: 0.9476 - val_loss: 0.2346 - val_accuracy: 0.9
```

Figure 11: CNN epochs

Since we are dealing with images, there is need to extract features from the images. A pre trained model VGG16 is used to do this.

```
In [23]: # load the model
         vgg_model = vgg16.VGG16(weights='imagenet')

         # remove the last layers in order to get features instead of predictions
         feat_extractor = Model(inputs=vgg_model.input, outputs=vgg_model.get_layer("fc2").output)

         # print the layers of the CNN
         feat_extractor.summary()

         Model: "model"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         input_1 (InputLayer)         [(None, 224, 224, 3)]     0
         _________________________________________________________________
         block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
         _________________________________________________________________
         block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
         _________________________________________________________________
         block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
         _________________________________________________________________
         block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
         _________________________________________________________________
         block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
         _________________________________________________________________
         block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
         _________________________________________________________________
         block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
         _________________________________________________________________
         block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
```

Figure 12: VGG16 model training

The next step is to comput the cosine similarity scores between the image features. Sklearn's metrics is used to perform this. The final step is to build the recommendation engine. A function is created that takes in the Product ID, Number of recommendation and returns N recommendations. The similarity score is arranged in descending order and results are outputed based on score.

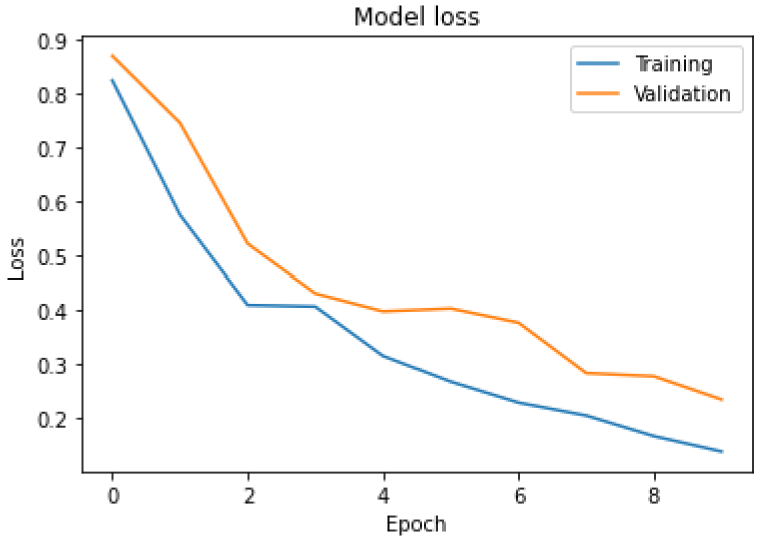
```
In [28]: # extract the images features

         imgs_features = feat_extractor.predict(processed_imgs)

         print("features successfully extracted!")
         imgs_features.shape

         features successfully extracted!

Out[28]: (1196, 4096)

In [29]: # compute cosine similarities between images

         cosSimilarities = cosine_similarity(imgs_features)

         # store the results into a pandas dataframe

         cos_similarities_df = pd.DataFrame(cosSimilarities, columns=files, index=files)
         cos_similarities_df

Out[29]:
```

	C:\Users\Admin\Desktop\seminar\images\\images0.jpeg	C:\Users\Admin\Desktop\seminar\images\\images1.jp
C:\Users\Admin\Desktop\seminar\images\\images0.jpeg	1.000000	0.8282
C:\Users\Admin\Desktop\seminar\images\\images1.jpeg	0.828224	1.0000

Figure 13: Feature extraction and cosine similarity

8 Results

8.1 Prediction accuracy for dataset.



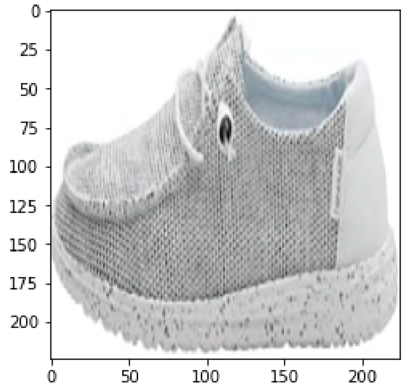

```
In [26]: print("Accuracy : ", cnn.evaluate(test_images, test_labels))

23/23 [==============================] - 49s 2s/step - loss: 0.2775 - accuracy: 0.8997
Accuracy :  [0.27745285630226135, 0.8997214436531067]
```

Figure 14: Model Accuracy

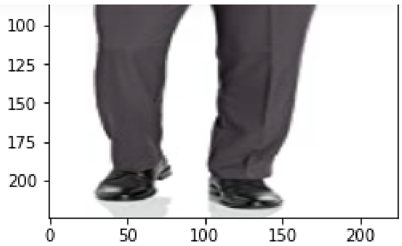
8.2 Implementation Result

8.2.1 Classification Result

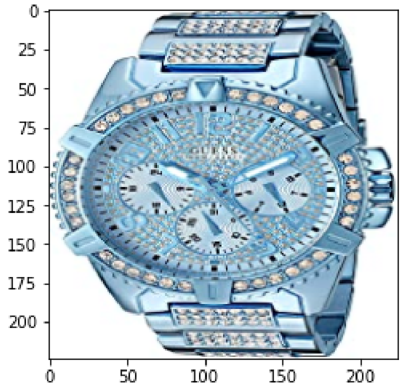
```
In [29]: get_image_and_class(5)
```




```
This is a Shoe!
```




```
This is a Pant!
```




```
This is a Watch!
```

Figure 15: Classification

8.2.2 Recommendation Result

Figure 16: Recommendation Result

As depicted in the above screenshots, 5 similar products are recommended based on the original/input image.

9 ADVANTAGES AND DISADVANTAGES

9.1 Advantages

With CNNs, it is able to extract the features of the image and using the suitable Machine Learning classifiers helped to achieve more accurate recommendations. Image searching is quick and easy. It offers simple and advanced search and also allows sorting options.

9.2 Disadvantages

Recommendation Systems usually need a lot of data in order to recommend appropriately. But they did not consider the problem of Cold Start. Low level features will not be able to interpret accurately.

10 APPLICATIONS

Image-based search has wide applications ranging from search engine optimization to product recommendation in e-commerce. Few of them are listed below-

1. Automated Image Organization – from Cloud Apps to Telecoms.
2. Stock Photography and Video Websites.
3. Visual Search for Improved Product Discoverability.
4. Image Classification for Websites with Large Visual Databases

11 CONCLUSION

A two phase recommendation model is built, which performs classification and recommendation in respective phases. Convulational Neural Network is used for classification purpose and similarity scoring is done using Cosine Similarity for recommendation phase. A recommendation model thus generates results with decent accuracy of 89.9%. In future the Recurrent Neural Network architecture can be used to merge non-visual data with visual data for recommendation process.

11.1 Scope

The product can successfully take images from the user as input and number of required recommended products needed, perform classfication task to determine the product category, and recommend the similar products by calculating similarity scores and fetching them from the dataset. In future the product would likely perform visual search on the user click movements, history, etc. to recommend products based on user's browsing activities.

11.2 Future Work

1. Build a recommendation engine that incorporates both the image-based and text-based methods.

2. Try to train our model on a larger amount of data using batches. This can potentially increase the accuracy of the model.

3. Create a web app to deploy the model- Classification Recommendation

References

[1] Farhan Ullah, Bofeng Zhang, and Rehan Ullah Khan, “Image-Based Service Recommendation System: A JPEG-Coefficient RFs Approach”,*IEEE, 2020*.

[2] Luyang Chen, Fan Yang and Heqing Yang, “Image-based Product Recommendation System with Convolutional Neural Networks.“, Stanford University cs213n reports, 2017.

[3] Hessel Tuinhof, Clemens Pirker and Markus Haltmeier, “Image Based Fashion Product Recommendation with Deep Learning ”, July 2018.

[4] Surbhi Jain and Joydip Dhar, “Image Based Search Engine Using Deep Learning.”, Proceedings of 2017 Tenth International Conference on Contemporary Computing (IC3), 10-12 August 2017.

12 Appendix

12.1 Log Report

12.2 Internship Letter

12.2.1 Internship 1

Nitor
Brilliance @ Work

Internship Completion Letter

To Whomsoever It May Concern

This is to certify that Bhavika Karale has served in our organization from in the capacity of Intern – Projects and Delivery.

Bhavika has successfully completed the internship as Intern – Project and Delivery.

She is being relieved from the services of the company on the closing hours of 31st Mar 2020.

During her tenure we found her sincere, diligent & committed.

We wish her all success in future endeavors.

For Nitor Infotech Pvt. Ltd

Rohini Wagh
AVP – Human Resources

Nitor Infotech Pvt.Ltd.
SEZ unit 2, Block Rhine,5th floor,"B" Wing, Plot 3A, Embassy Tech Zone,
Rajiv Gandhi Infotech Park, Phase II, Hinjewadi, Pune- 411057
www.nitorinfotech.com | info@nitorinfotech.com

Figure 17: Internship Letter- 1

Report of Internship Project

1. Worked as a Data Analyst in a team of four for a healthcare company client
2. Worked on cleaning the data set and explored data pattern through Data Visualization techniques

12.2.2 Internship 2

ATOMIC loops

Email: info@atomicloops.com
Website: www.atomicloops.com
linkedin.com/company/atomicloops

Date: 01 February, 2021

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. Bhavika Karale** has successfully completed her internship in Digital Marketing at Atomic Loops from 08th August 2020 to 01th February 2021.

During the internship she demonstrated good development skills with a self-motivated attitude to learn new things. Her performance exceeded expectations and was able to cope up with new things in a very short time. She was hardworking, dedicated, and committed. It was a pleasure having her with us in this period.

Her association with us was really fruitful and we wish her all the best for his future endeavours.

Sincerely,
The Admin Team,
Atomic Loops

Figure 18: Internship Letter- 2

Report of Internship Project

1. Developed a strategy for wider reach among the audience
2. Learned to create and deploy ads on social media platforms
3. Handled social media accounts, created technical content