Design and implementation of clothing fashion style recommendation system using deep learning

Muhammad KHALID[1], Mao KEMING[1], Tariq HUSSAIN[2]

neuse202@gmail.com, maokm@mail.neu.edu.cn, uom.tariq@gmail.com

[1] Northeastern University Shenyang Liaoning, China

[2] School of Computer Science and Information Engineering,
Zhejiang Gongshang University Hangzhou, China

Corresponding Author:
Tariq HUSSAIN
uom.tariq@gmail.com, 20010109003@pop.zjgsu.edu.cn

Abstract: In recent years, the huge amount of information and users of the internet service, it is hard to know quickly and accurately what the user wants. This phenomenon leads to an extremely low utilization of information, also known as the information overload problem. Traditionally, keywords are used to retrieve images, but such methods require a lot of annotations on the image data, which will lead to serious problems such as inconsistent, inaccurate, and incomplete descriptions, and a huge amount of work. To solve this problem, Content Based Information Retrieval (CBIR) has gradually become a research hotspot. CBIR retrieves picture objects based entirely on the content. The content of an image needs to be represented by features that represent its uniqueness. Basically, any picture object can be represented by its specific shapes, colors, and textures. These visual characteristics of the image are used as input conditions for the query system, and a result the system will recommended nearest images and data set. This research designs and implements two-stage deep learning-based model that recommends a clothing fashion style. This model can use deep learning approach to extract various attributes from images with clothes to learn the user's clothing style and preferences. These attributes are provided to the correspondence model to retrieve the contiguous related images for recommendation. Based on data-driven, this thesis uses convolutional neural network as a visual extractor of image objects. This experimental model shows and achieves better results than the ones of the previous schemes.

Keywords: Cloth Recommendation, Convolutional Neural Network, Similarity Measure.

1. Introduction

During the last years, online shopping has been growing. In 2013, the total turnover for e-commerce in Europe expanded with 17% in contrast to the 12 months before and huge organizations can have hundreds and hundreds of products or even more from which we can select on websites. Both the customer and the business enterprise desire the client to easily discover applicable products or items both throughout search and when they are searching, and this is where recommender systems come into the picture [1]. The greater part (62%) of US buyers with Web access presently shop on-line, to some degree, at least a month, and 1% say they do not buy from internet, as indicated by a current report by Walker Sands [2] .From all the clients looking for items on the web, 63% of them buy garments (Burke, 2002), these being, quite possibly, the most purchased items. The information uncover that women are more likely to buy on-line, with 71% of ladies doing this, contrasted with 52% of men (Reshma & Patil, 2012). Studies on clothing are in a growing development in general as a result of the tremendous market related to dress. In China, the serviceable market crushed 20 billion US dollars in 2016 (Jannach & Friedrich, 2013). Such huge market prospects impressively energize clothing applicable exploration. Being one of the new studies in progress both at national and international level, recommender systems have proved to be a large solution for e-commerce (Beel et al., 2013), but the internet options yet pose many strong and weak points. Some of these weaknesses consist in lack of accuracy regarding information, which is the more important weaknesses amongst others (Massa & Bhattacharjee, 2004). To decrease some of these weaknesses, collaborative filtering methods have been combined with content-based methods to come up with hybrid recommender systems (Massa & Bhattacharjee, 2004). Moreover, explicit and implicit remarks have also been mixed to enhance the accuracy of recommenders (Massa & Bhattacharjee, 2004; Guo et al., 2014). Absence of precision is basically

https://doi.org/10.33436/v31i4y202110

because of errors coming with the use of contradicting algorithms, incapable to realize contrasting issues between have distrust and faith, putting into consideration the web of have faith (Massa & Bhattacharjee, 2004; Abadi et al., 2016). Picture recovery can be depicted as the errand of looking out for pics in a picture data set. This is present not an astute thought, in light of everything. It has been explored on account of the way that the 1970s joined informational collection associations with PC vision, looking into the issue as indicated through two uncommon perspectives, the first being text-based and the second one being visual-based. From the outset, the developments have been made only through information annotations that have been saved in a database to work the retrieval step, however, when the dimension of the image collections started to amplify the effort required to label them used to be as soon as unsustainable, to solve this issue, during the 1990s, content-based photograph retrieval was proposed. Starting now many searched for lines have seemed the use of one or the different isolated or combining them. (Reshma & Patil, 2012) (Alkhawlani et al., 2015).

From previous analysis implemented until now and from the exploitation of various research algorithms and unique experiments, the recommender system is still the most appropriate answer for solving future net issues. This is considered an important study at both national and international level. The flexibility of these systems was rapidly enhanced by the collaboration between the local researchers and the international ones. Recommendation systems make recommendations based on the information they are provided with and in the manner in which they are programmed. Going into details, most of the evaluation applied is independent coming up with a brand-new recommendation algorithm, system, or model. However, different researchers use already existing work as researchers use an already existing current piece of work to come up with a new diagram or to truly improve the current one. The present analysis model focuses on the use of a current algorithmic program and, consequently, the use of a new research concept comes up with a recommender system. Existing research and fashions have given us some inspirations of how to design fashion recommendation systems. Nevertheless, they also involve some common drawbacks. Therefore, in this study, our aim to suggest a new method to assist personal choice making through supplying images and get suggestions based on provided contents. The contribution of the research are follows:

- To design and implement a web-based clothing fashion style recommender system based on deep learning;
- A scheme for improving a person's clothing style by removing the features he/she doesn't like;
- from his/her clothing images;
- These attributes served to a similar model to retrieve similar images as recommendations;
- Combined with more common content-based recommendation systems, our model can help to extend robustness and performance, for example can suit a more pretentious style of a client.

Section 2 looks at the related work from the specialized literature. Section 3 analyses the architecture of the proposed system. Section 4 presents and evaluates the experimental results and Section 5 offers the conclusion.

2. Related work

Online business has been set up for around at least 30 years (Mirescu & Maiorescu, 2010; Tian & Stewart, 2008). The on-line media has extraordinarily influenced the whole lifestyle worldwide or, at least, 99% of it. Everything started in the late 80's, when the web entered the lives of many individuals around the world. Regardless of the numerous issues it brought about in the mid 2000's, the internet business industry has developed quickly and impacted all sides of employment in the public area. There has been an expanded pace of improvement in the public eye due to internet business (Mirescu & Maiorescu, 2010).

Recommendation systems have been described via many researchers in extraordinary ways. Some have described them as supporting systems which assist customers discover data about products for their pastime quicker than if it weren't for them (Park et al., 2011). They can additionally be defined as software programs which help users to decide and predict their wishes by analyzing the consumers' behavior and their shopping records (Jannach & Friedrich, 2013). They can additionally be defined as a statistics filtering techniques that are able to supply guidelines of commercial items for customers (Lee, 2012; Bobadilla et al., 2011). The goal of recommendation systems is not to make money. However, some groups like Amazon have turned this into a money-making commercial enterprise as it helps improving their sales. Building recommendations has also resulted into a robust enterprise as it works with maintaining loyal customers (Claypool et al., 1999). Recommender systems no longer consist only in calculating consumers' similarities to make recommendations, but also in analysing in detail the consumer' s trends and in mining facts (Claypool et al., 1999). Since The adoption and introduction of an improved e-commerce in the last 30 years (Tian & Stewart, 2008; Mirescu & Maiorescu, 2010) has come with many problems which have impacted the operation of the recommendation systems. These websites encompass many customers and the availability of too much information. Browsing from a webpage to another, shopping online based on many alternatives and a great deal of detail have been a setback for e-commerce. Despite all these problems, humans cannot renounce to use it. The more people use it, the more information is generated. Customers tend to discover it challenging to get access to the required and efficient information (Isinkaye et al., 2015). In order to resolve this problem, e-commerce providers and shops have resorted to the use of recommendation systems. These are intelligent systems that can depend upon a single mouse click or key stroke to study the conduct of costumers and predict what their desires are (Krizhevsky et al., 2017) (Melville & Sindhwani, 2010). Pazzani's method (Pazzani, 1999) makes use of a person's profile as a vector illustration of weighted words derived from positive training example, by employing the Winnow algorithm. Several hybrid technics are considered to be the classification tasks (Pine, 1993, Schafer et al., 1999).

3. Proposed system architecture

The system architecture defines the hardware, software and network environment of the structure. The system will be web-based meaning that the users need to run the URL in order to run the system. The system will run both horizontally and vertically. The architecture used in the system is shown horizontally where the Model View Controller is explained as represented in Figure 1. The high-level part of the system is looked at using the vertical way.

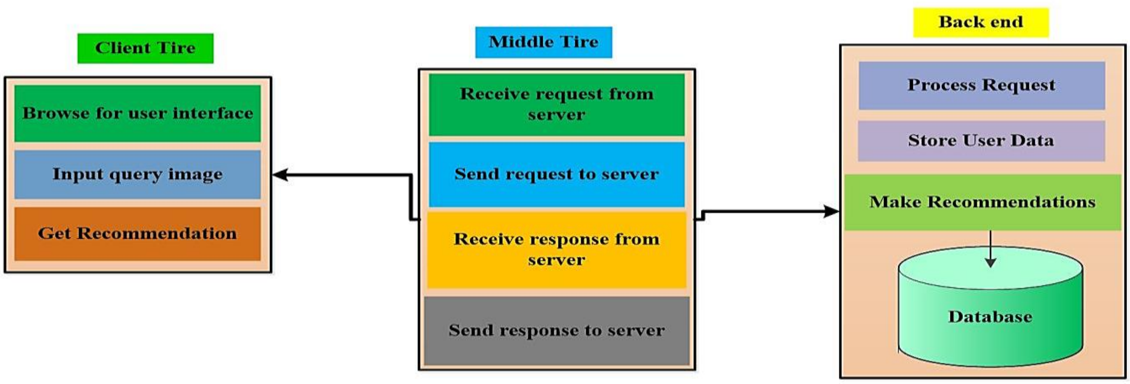

Figure 1. Three tire system architecture

The system comprises of the Client tire, which is the front end or View mode, middle tier which is the system controller and the backend tire which is the model. The client side is where the users/customers log in in the system, browse for the system interface, provide input query image to the system, and get recommendation according to the input query. The middle true is responsible for communication between the front end and the back end. It receives user requests and sends them to the back end and in turn accepts responses from the back end and sends them to the user.

The back end which involves the data set and recommender algorithm deals with data storage, user input data storage, processing user requests, determining user input similarity, making recommendations and forwarding them to the middle tier which in turn sends them to the respective users. The internet works to provide access to the site with a strong security check, provided by both firewall and password protection policy. Any unauthorized access is detected and prevented by the firewall.

a. The vertical classification system model

In Figure 2, the recommendation system works with the data set to track user input data features and extracted features from data set upon which new predictions and recommendations are made. the recommendation browses the dataset for user data and available dataset features.

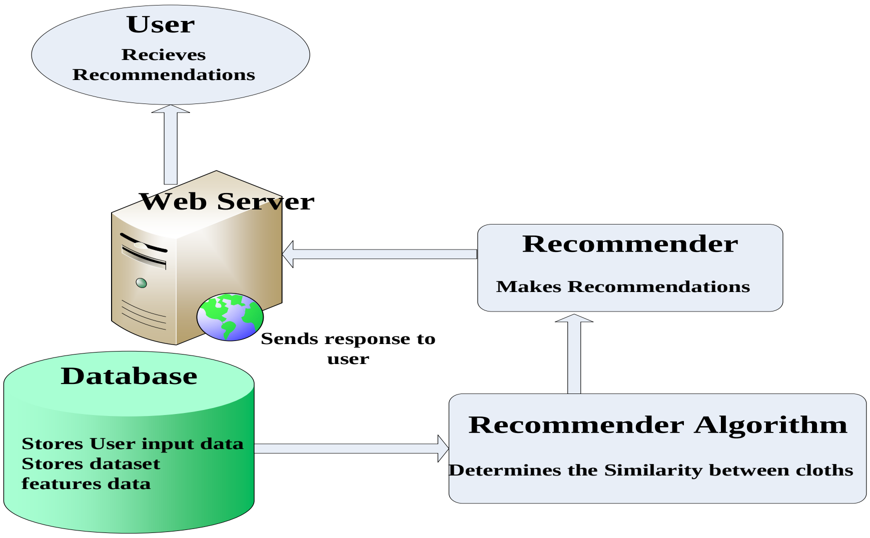

Figure 2. Vertical architecture of the system

Then, it uses the algorithm to go over the input user data and determine similarities between users input data and stored dataset features. Finally, it makes recommendations. By looking at Figure 1 and Figure 2, we realize that the recommender system does not interact directly with the users at any point.

When the repository stores data, the recommender filters the data it needs from the repository using the algorithm. When a signal is sent to the algorithm about what data are needed for filtering, the algorithm computes the similarity. The similarity results are then transferred to the recommender system which in turn sends recommendations to the webserver and finally to the respective user.

b. Dataset and classification

In this project, we worked with the Deep Fashion dataset, which is gathered from researchers from the Chinese Hong Kong University. It has over one million diverse trend pics and wealthy annotations with additional data about landmarks, categories, pairs etc. The dataset consists of 5 distinct types of predicting subsets that are tailor-made towards their tasks.

Figure 3. Fashion dataset

One subset, known as Attribute Prediction, can be used for apparel category and attribute prediction. From almost 290,000 photos of 50 apparel categories and 1,000 apparel attributes, we randomly picked 18k images from different categories and then we classified them for training and testing. The distribution of labels is presented in Figure 3.

c. Design of deep learning module

There are many classification algorithms or classifiers in use today. The most notably and the most implemented classifiers are Vgg-16, Vgg-19, AlexNet, BN-Inception, ResNet etc. In our system, the Vgg-16 classifier and feature extractor are implemented to solve a problem of cloth / fashion recommendation process.

$$\mathbf{N_i(Wi,.) \triangleq (Vi, Fi, (Ui,.)) for\, i = 1, 2} \tag{1}$$

Here $w_i = (u_i,v_i)$ are weight vectors, $s_i(v_i,.)$ are fully connected softmax output layers that actually perform classification and $f_i(u_i,.)$ are the CNN without the last layer. They are used as a feature extractors.

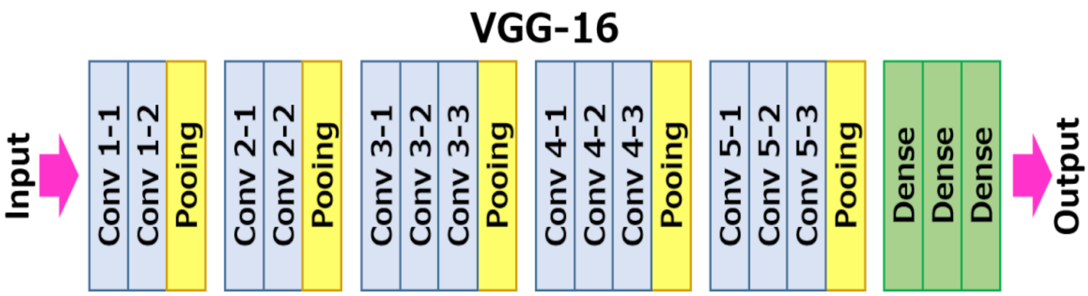

Figure 4. VGG16- Architecture

The core network of our model is VGG16 as shown in Figure 4. VGG16 was projected by Simonyan, K. and Zisserman, A. who presented a convolutional neural network in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition", at the University of Oxford.". Then model is checked for top-5 accuracy on ImageNet. which contains 14 million images datasets belonging to one thousand classes and achieves the value of 92.7.

d. Design of visual recommendation module

The fashion domain is a very popular playground of machine learning and computer vision. The main problem of this domain is produced by the high level of subjectivity and the semantic complexity of the features involved. Recent work has focused on a variety of approaches including attribute recognition, clothing retrieval, image generation and visual recommendation. Table 1 shows different distance measurement formulas for image feature vector similarity, and definitions for the mentioned similarity measures as presented by (Iglesias & Kastner, 2013).

Table 1. Distance measures available for image feature vector similarity

Name	Formula	Description
Manhattan (City Block)	$\mathbf{d_{CB} = \sum_{i-1}^{d} \| P_i - Q_i \|}$	Some of the absolute values of two points. (L1 Minkowski order 1)
Euclidean	$\mathbf{d_{Euc} = \sqrt{\sum_{i-1}^{d} \| P_i - Q_i \|^2}}$	Square root of the sum of the squared distances between points. (L2 Minkowski order 2)
Chebyshev	$\mathbf{d_{Cheb} = \max_i \| P_i - Q_i \|}$	Max distance between points along any coordinate dimension. ($L\infty$ Minkowski order ∞)
Hammington	Compare the first two bits in each string. If same. Record a “0”, else “1”	Sum of differences between two binary strings.
Cosine	$\mathbf{S_{Cos} = \frac{\sum_{i-1}^{d} \| P_i - Q_i \|}{\sqrt{\sum_{i-1}^{d} P_i^2} \sqrt{\sum_{i-1}^{d} PQ_i^2}}}$	Measures the cosine of the angle between two vector points.

First closeness measure between style dataset and client input First, we need to build a style profile for the client, which is then refined through taking at least one of the client’s photographs from his/her ideal clothing objects. Then, the design vectors are entered and fostered. These vectors are then joined to shape the framework of the style profile for each individual. The component vectors prepare the information from the dataset.

- **set** train data **= get**
- **for each** model **in the list (**Vgg16) **do**
  - **for each** distance metric **in (**Similarity**) do**
  - **train** new **nn** model **(**train data**,** distance metric**,** neighbors=5**)**

Presently we utilize a similitude calculation to assess the design vector of each image in the archive with the style profile lattice. This offers us a score dependent on the wide assortment of component matches - the higher the score the nearer an image is to the individual's style profile. At that point, we rank the photos arranged in their classification and show, as proposals, the pictures with the highest rank.

set test data **= get** feature vectors of the test data from the repository database
- **for each** model **in range(n) do**
  - **for each** image **in** test_data **do**
    - **extract** neighbors top five from model
    - **store** results in the database

4. Experimental results and evaluation

This section focuses on evaluating our system and deciding the stage to which it is able to fulfill the purpose for which it was created the performance of the system is analysed in detail through several tests, from small scale to large scale. Firstly, the unit tests are done at the lower stages and then we proceed to the whole test system. Several machine purposes are also involved in the system. In the training implementation module, we are performing the movement throughout the area, freeze the base layers of the organization i.e., the VGG16 layers, and train the model on the dataset for 5 epochs. This trains the external layers to figure out how to characterize the pictures. We then unfreeze the lower layers and train the model for 5-7 epochs until the approval exactness settles. We keep the best achievable loads (best on approval exactness) and use it for the suggestion model. The training implementation code is presented below.

Step 1: *Training the whole network for 5 epochs first*
Step 2: Checkpoint_callback=modelcheckpoint(‘, /model/vgg_weights_best_pattern.hdf5’.
Step 3: Monitor=’val_acc’, verbose=0 save_best_only=true, save_weights_only=false, mode=auto’, period=1)
Step 4: Tf_model.fit_generator(
Train_generator,
Samples_per_epoch=nb_train_sample,
Nb_epoch=10,
Validation_data=validation_generator,
Nb_val_samples=nb_validation_samples,
Verbose=1,
Initial_epoch=5,
Callbacks=[checkpoint_callback]
Step 5: end

4.1. Visual recommendation module implementation

To get proposals, we wished to construct a vault of pictures. This archive would be a unique application. If the suggestion was cultivated for shopping, the storehouse would have contained pictures from online retail locations like Amazon, eBay, Pinterest, Instagram, etc. A subset of pattern datasets was used to test our proposed approach. At that point, the information had already been cleared of unimportant photos. Then, the photos were passed by means of the organization and design vector pictures have been created from each photo. For the getting the suggestion, we first needed to build the individual style profile. This is brought out by taking one or more noteworthy pictures of the client's ideal attire things as they were entered and by making their style vector. These vectors are then blended to shape the framework of the individual style profile. The Figure 5 shown Pattern recommendation with similarity score.

Figure 5. Pattern recommendation with similarity score

The proposed scheme is further below, as follows: we will utilize a closeness calculation, which analyzes the design vector of each picture in the vault with the style profile grid. This gives us a score dependent on the quantity of component coordinates (i.e., how great is the degree of similarity of a picture to the individual's style profile).

```
Step 1: def similarity (feature_data, inp_ feature_data):
Step 2: nun_samp=inp_feature_data. size
Step 3: print (unm_samp)
                    Sim_score = []
                    for i in range (1 en (feature_data)):
                    score=0
Step 4: show_sample (data_images[i])
Step 5: print (feature_data[i])
                    score _m = inp_feature_data - feature_data[i]
Step 6: print (Soore_ m)
                    score= nun_samp-np. Count_nonzero (score_m)
                    sim_score [i]=score
Step 7: print (score)
                     sim_Score
Step 8: end
```

i. User management services

The system provides a platform through which a user can visit the system and provide his/her choices regarding the fashion images for best recommendation.

ii. Fashion vector for images in repository and input fashion vector

The system is responsible for making fashion vectors for images in the repository and fashion vector images provided by the user to the system, for the similarity measures and for making recommendations. After making the fashion vector, some predictions are made, as illustrated in Figure 6.

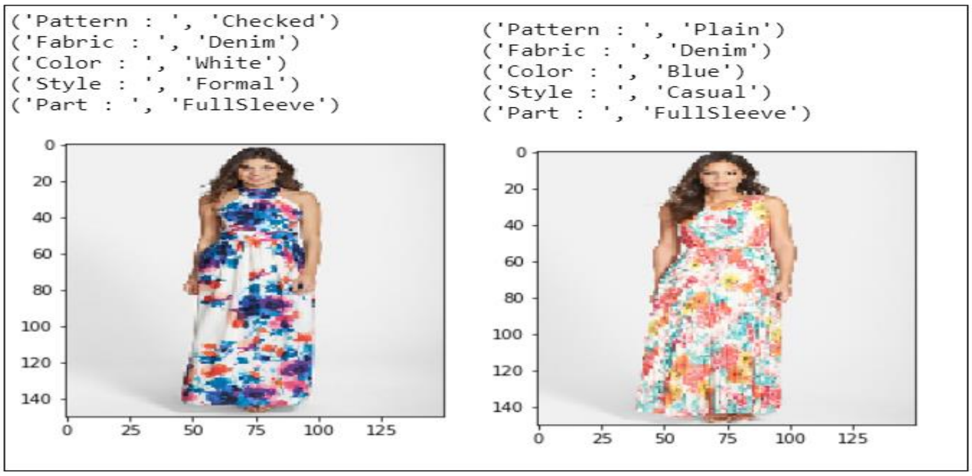

Figure 6. Input images with predicted values

iii. Recommender service

The system is responsible for making recommendations to users based on their user data. The user data compiled in the dataset is filtered by the recommender system through the recommender algorithm.

```
Step 1: Def similarity (feature_data, inp_feature_data);
          Num_samp=inp_feature_data.size
Step 2:       print (num_samp)
           Sim_score = ()
           For i in range (len (feature_data));
              score = 0
                   show_sample (data_images[i])
                   print(feature_data[i])
```

```
Step 3:          Score_m inp_feature_data-feature_data[i]
                 print (score_m)
Step 4:          Score=num_samp-np.count_nonzero(score_m)
            Sim_score[i]=score
                print(score)
Step 5:     Return sim_score
```

The recommender algorithm is able to calculate and determine similarity between the user inputs by utilizing their available extracted feature data and by presenting the results to the recommender system which in turn makes content-based recommendations to users.

```
Step 1: Similarities=similarity(feature_data,inp_feature_data)
Step 2: Sorted_similarties=sortesd(similarities.items(), key=operator.itemegtter(1), reverse=true)
        print (sorted_similarities)
                Num_reco=30
                        Num_data=feature_data.size
                For I in range(num_reco)
                Ind = sorted_similarities[i][0]
                Print ("score:", sorted_similarties[i][1])
                        Show_sample(data_images[ind])
Step 3: end
```

By accessing the system, users are able to access and view their content-based recommendations. However, all the recommendations are made based on the similarity between user inputs and user inputs. As long as there is a level of similarity, we make the best recommendations.

iv. Recommender to the query images in dataset

We can see that our model can capture the best matching style by including the length, shape, color, fabric and pattern of the cloths, as illustrated in three query images examples. In the first example, the model captures deep features including the blouse category, fabric, repeated floral pattern and the regular fit style. As seen, the five recommended images display different clothes. The second example shows that the model captures the wool fabrics, the contrast color stitches and the turtleneck. The third example shows that the model can capture the cotton fabrics and the printed letters. The recommendations can be seen in Figure 7.

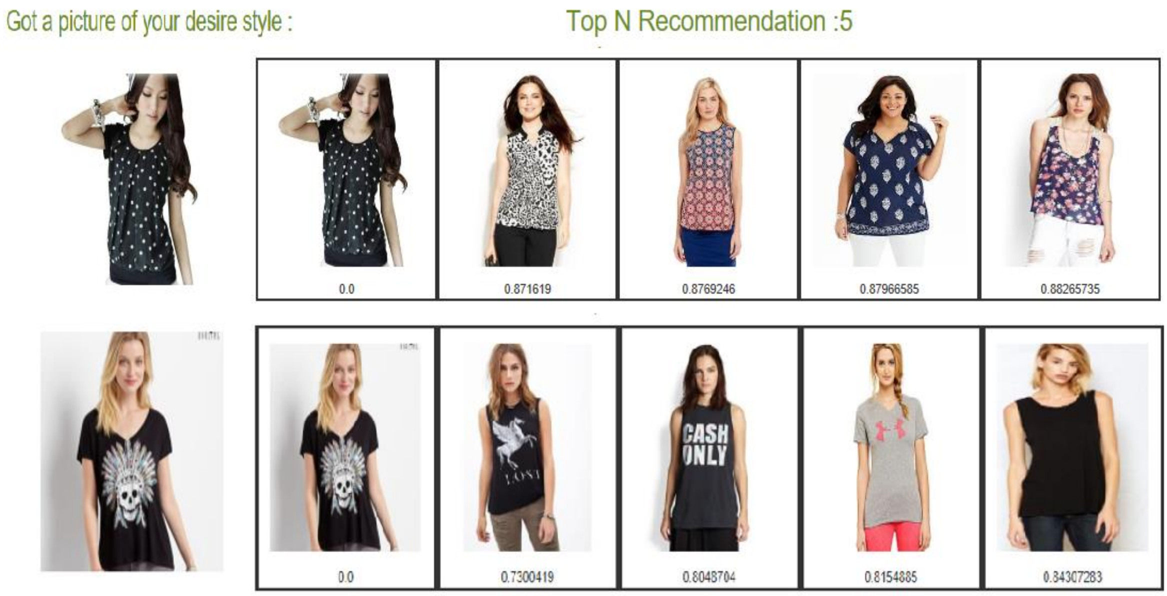

Figure 7. Pattern recommendation

As shown in Figure 7, our model can capture the style with high accuracy, meaning that our system achieves its purpose. It can be noticed that our system can perform for all the involved categories like pattern, style, fabric etc. The highest similarity score shows that the input images and the recommended ones are similar. This figure also illustrates that the system can work best for pattern recommendation and recommend top similar images in different colors, shapes, and styles.

v. Recommendations to the query images outside the dataset

It's natural to ask if the model you made works with images which are not part of the dataset. We randomly downloaded three online images illustrating expensive clothes. As shown in Figure 8, the model is still able to capture the style, pattern and fabrics of the clothes and recommend similar ones.

The model is checked for different categories like pattern, style, fabric. The highest score show that the image is more similar to the input query. So, our model obtains high similarity score for different categories, as shown in Figure 8.

Figure 8. Outside recommendation dataset

4.2. System result and accuracy

Finally, this subsection evaluates the system and shows the testing results and the accuracy of our model. After adding the model on top of the convolutional base, freezing the weights of all layers except of the top ones, and training the model for 5 epochs, the following accuracy was obtained, as shown in Figure 9.

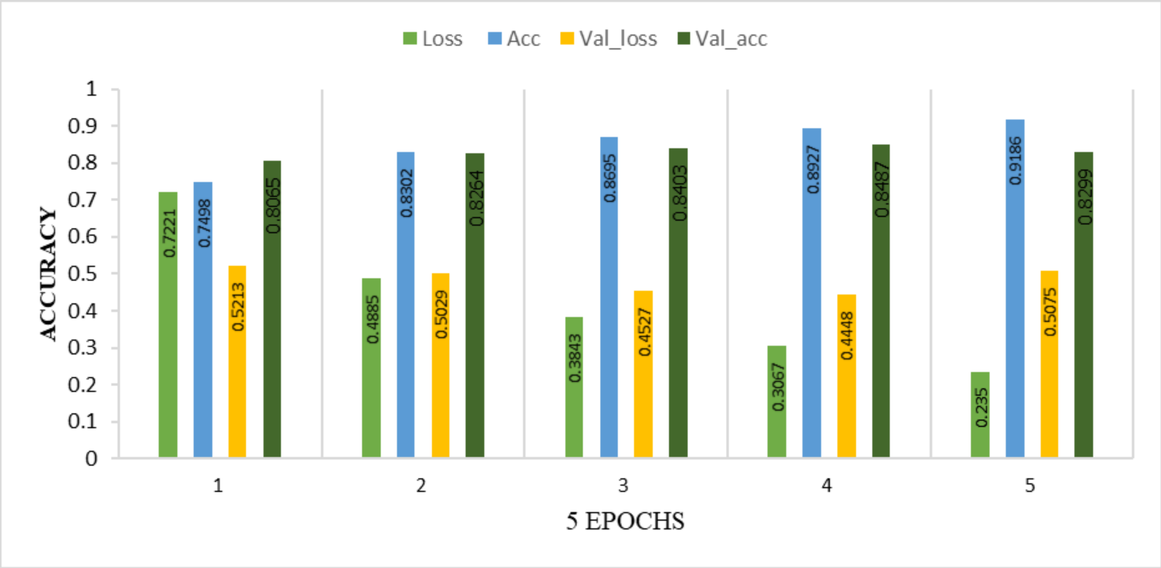

Figure 9. Model accuracy after freezing the layers for 5 epochs

After calculating the mean accuracy for 5 epochs, the obtained results are as follows:

Validation: accuracy = 0.836000; loss = 0.489109

This part of the sentence “After calculating the mean accuracy for 5 epochs” is mentioned also below, after Figure 10, and these values mentioned for accuracy and loss (0.836000 and 0.489109) are not illustrated in Figure 9, but in Figure 11.

After that unfreeze the layers and train model for 5 ages to get the end-product of our model which is further used in Figure 10.

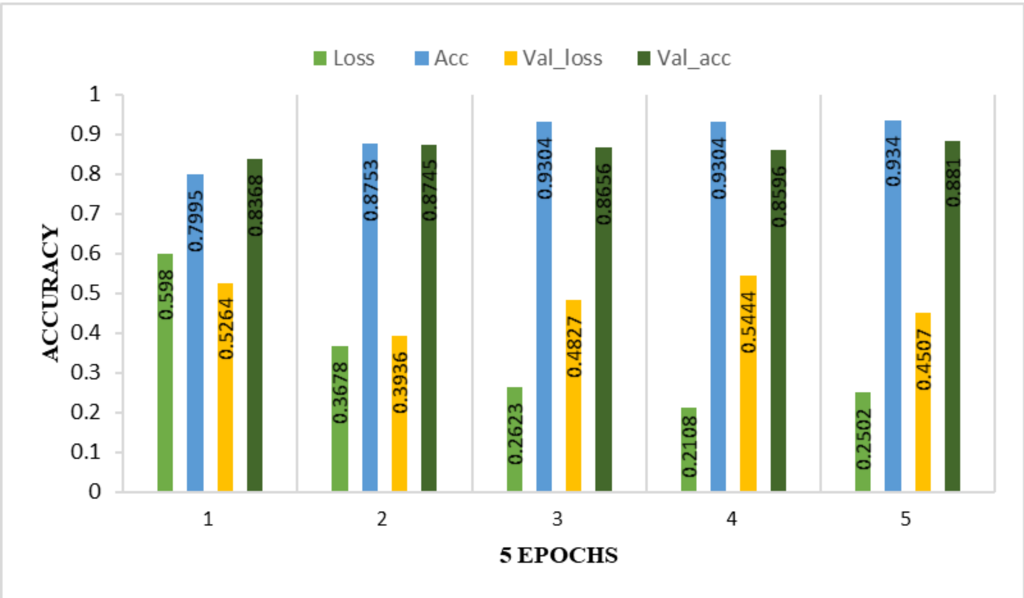

Figure 10. Model accuracy for 5 epochs

After calculating the mean accuracy for 5 epochs, the final result are as follows:

Validation: accuracy = 0.864750; loss = 0.516400

These values mentioned for accuracy and loss (0.864750 and 0.516400) are not illustrated in Figure 10.

The accuracy of our model was compared with the one of Alex Net model. It can be clearly noticed that our model gives a better accuracy when compared to Alex Net, as shown in Figure 11.

Figure 11. Accuracy and loss

Recommender systems are still developing and, as extra research is being done, extra areas and weaknesses that need greater study are also developing. Recommender systems have proved to be a great solution to the overload of web data, an important problem affecting the users. With the ever-growing records and choices, recommender systems enable the customers to access the data they need within minutes, just by a mere mouse click or by a single key stroke. Table 2 shows the comparison with other models regarding the accuracy and the loss values.

Table 2. Comparison with existing models

Model	Accuracy	Loss
VGG 16	0.83	0.48
AlexNet	0.79	0.51
Our Model	0.86	0.51

5. Conclusion

The present paper presents the development of a system that recognizes fashion similar images. We accomplish this by implementing an already existing CNN model with transfer learning for cloth image recognition using different libraries. For this purpose, we created a plan for collecting data and for developing the steps needed for preprocessing and cleaning up the data. We took into account features like patterns, machine, fabric, style etc. After extensive preprocessing and cleaning of data in a dataset, we constructed the model of stacked CNN to predict the features specific to these attributes and to train the models with the dataset to generate accurate predictions regarding almost all forms of images. A stacked CNN was used and implemented, with the help of this algorithm through which the system can recommend similar images This is the last test to assess if deep learning for style recovery is at a high development and can be utilized in making fashion choices.

REFERENCES

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J. & Devin, M. (2016). *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*, pp. 265–283. arXiv preprint arXiv:1603.04467.
2. Alkhawlani, M., Elmogy, M. & EL Bakry, H. (2015). *Text-based, content-based, and semantic-based image retrievals: a survey*. Int. J. Comput. Inf. Technol, 4, pp. 58-66.
3. Beel, J., Langer, S., Genzmehr, M., Gipp, B., Breitinger, C. & Nürnberger (2013). A. *Research paper recommender system evaluation: a quantitative literature survey*. Proceedings of the

International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, 2013, pp. 15-22.

4. Bobadilla, J., Hernando, A., Ortega, F. & Bernal, J. (2011). *A framework for collaborative filtering recommender systems*. Expert Systems with Applications, 38, pp. 14609-14623.
5. Burke, R. (2002). *Interactive critiquing forcatalog navigation in e-commerce*. Artificial Intelligence Review, 18, pp. 245-267.
6. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. & Sartin, M. (1999). *Combing content-based and collaborative filters in an online newspaper.* Proc. of Workshop on Recommender Systems-Implementation and Evaluation, 1999, pp. 6-22.
7. Guo, G., Zhang, J. & Thalmann, D. (2014). *Merging trust in collaborative filtering to alleviate data sparsity and cold start*. Knowledge-Based Systems, 57, pp. 57-68.
8. Iglesias, F. & Kastner, W. (2013). *Analysis of similarity measures in times series clustering for the discovery of building energy patterns*. Energies, 6(2), pp. 579-597.
9. Isinkaye, F. O., Folajimi, Y. & Ojokoh, B. A. (2015). *Recommendation systems: Principles, methods and evaluation.* Egyptian informatics journal, 16, pp. 261-273.
10. Jannach, D. & Friedrich, G. (2013). *Tutorial: recommender systems*. International Joint Conference on Artificial Intelligence Beijing, 2013, pp. 1-144.
11. Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2017). *ImageNet classification with deep convolutional neural networks*. Communications of the ACM, 60, pp. 84-90.
12. Lee, J. S. (2012). *Survey of Recommender Systems*. Citeseer.
13. Massa, P. & Bhattacharjee, B. (2004). *Using trust in recommender systems: an experimental analysis*. International conference on trust management, 2004. Springer, Berlin, Heidelberg, pp. 221-235.
14. Melville, P. & Sindhwani, V. (2010). *Recommender systems*. Encyclopedia of machine learning, vol. 1, pp. 829-838.
15. Mirescu, S. V. & Maiorescu, T. (2010). The premises and the evolution of electronic commerce. Journal of knowledge management, economics and information technology, vol.1(1), pp. 44-56.
16. Park, D. H., Choi, I. Y., Kim, H. K. & Kim, J. K. (2011). *A review and classification of recommender systems research*. International Proceedings of Economics Development & Research, 5 (1), p. 290.
17. Pazzani, M. J. (1999). *A framework for collaborative, content-based and demographic filtering*. Artificial intelligence review, vol. 13, pp. 393-408.
18. Pine, I. (1993). BJ (1993). *Mass customization: The new frontier in business competition*, Vol. 17, issue 2, pp. 271-283.
19. Reshma, C. & Patil, A. (2012). *Content based image retrieval using color and shape features*. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 1, pp. 386-392.
20. Schafer, J. B., Konstan, J. & Riedl, J. (1999). *Recommender systems in e-commerce.* Proceedings of the 1st ACM conference on Electronic commerce, 1999, pp. 158-166.
21. Tian, Y. & Stewart, C. (2008). *History of e-commerce. Electronic commerce: concepts, methodologies, tools, and applications*. IGI Global. vol. 4, pp. 1-8.

Muhammad KHALID received his B.S. Degree from the University of Peshawar, in 2014, and the M.S. Degree from the Northeastern University, Shenyang China, in 2020. He is currently working as a visiting lecturer at Government Postgraduate College (GPGC). His research interests include Deep Learning, Computer Vision and Information Retrieval.

Mao KEMING received the B.S. and Ph.D. Degrees from the Northeastern University, Shengyang, China, in 2003 and 2009, respectively. He is currently an Associate Professor at the Northeastern University. His research interests include Computer Vision, Deep Learning and Intelligence Data Analysis.

Tariq HUSSAIN received his B. S. and M. S. Degrees in Information Technology from the University of Malakand, Pakistan (2015) and from the Institute of Computer Sciences and Information Technology – the University of Agriculture Peshawar, Pakistan (2019), respectively. He has published many research papers in the area of Computer Networks. He is currently a doctoral candidate at School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. His research interests are: Internet of Things, Big Data, Data Analytics, 3D Point Cloud, Joint Source and Channel Coding and Artificial Intelligence.