Team ID: PNT2022TMID03176

Project Name: SMART FASHION RECOMMENDER APPLICATION

Team Leader: Saravanan M

Team Member 1: Sathyan A

Team Member 2: Sedhu Vikram M

Team Member 3: Senthilvel K

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.

Introduction

During the last years, online shopping has been growing. 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. 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:

- 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.

2. 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 2. The high-level part of the system is looked at using the vertical way.

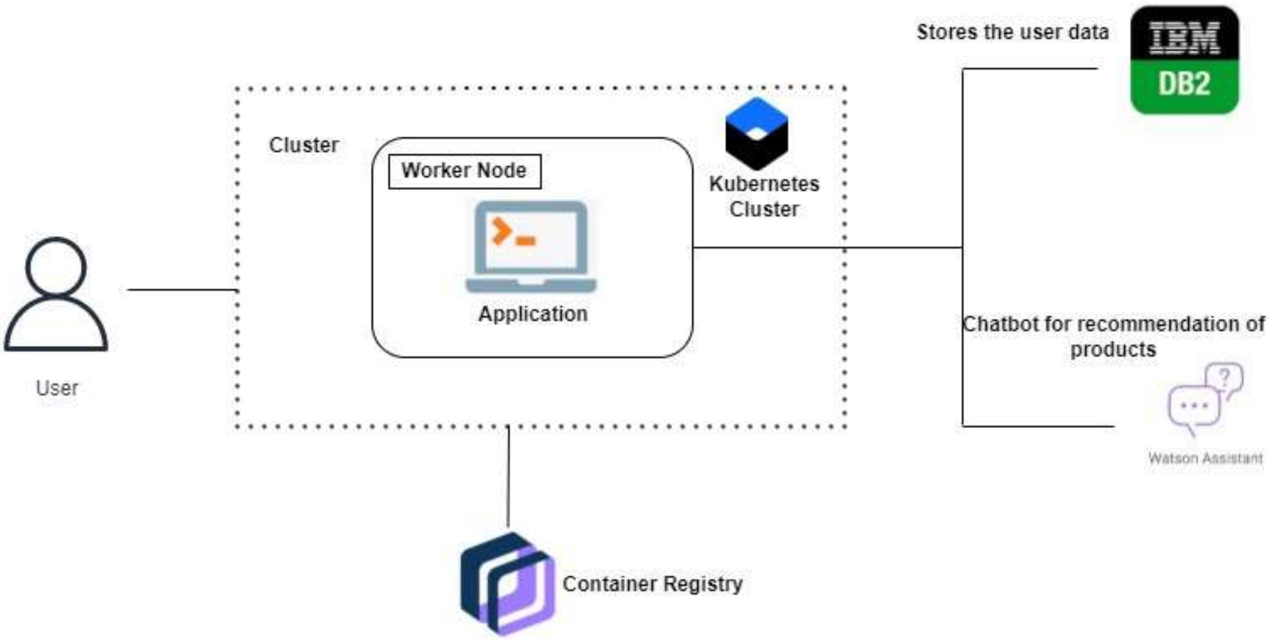

Figure 1. Technical Architecture

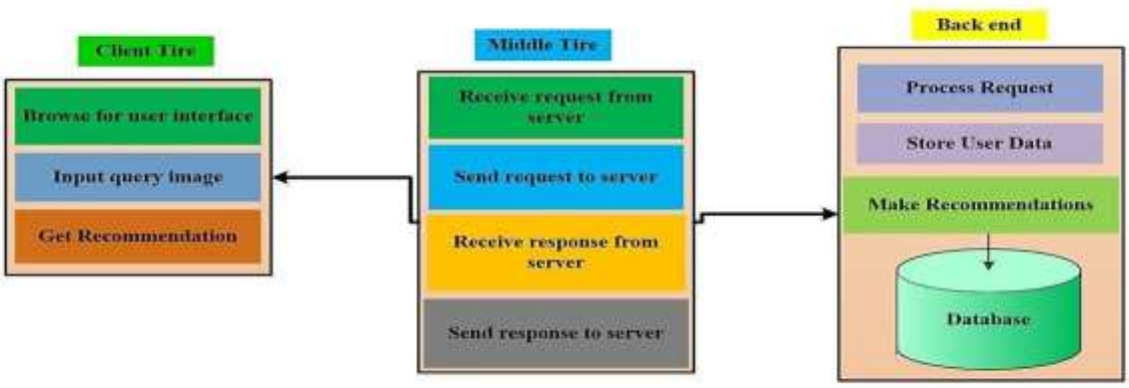

Figure 2. Three tire system architecture

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.

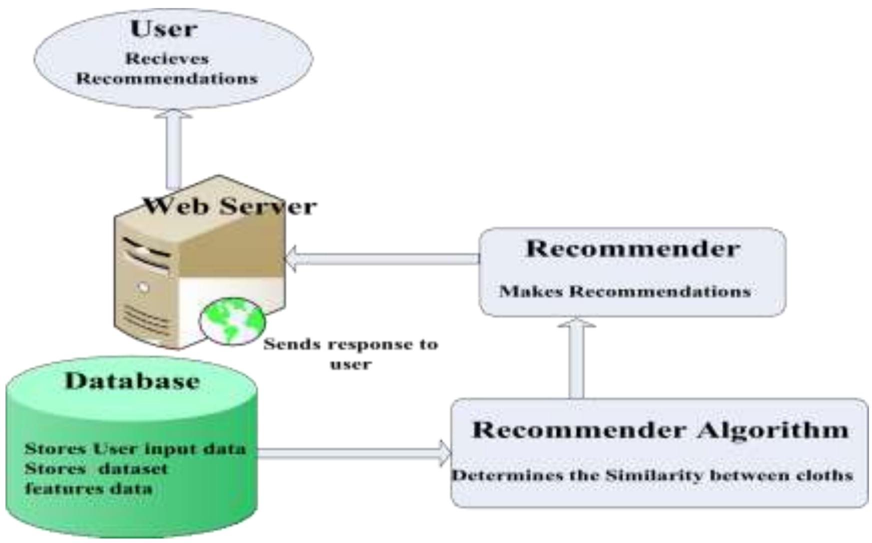

Figure 3. Vertical architecture of the system

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. 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 4. Fashion dataset

c. 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). 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 analyzed 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.

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).

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.

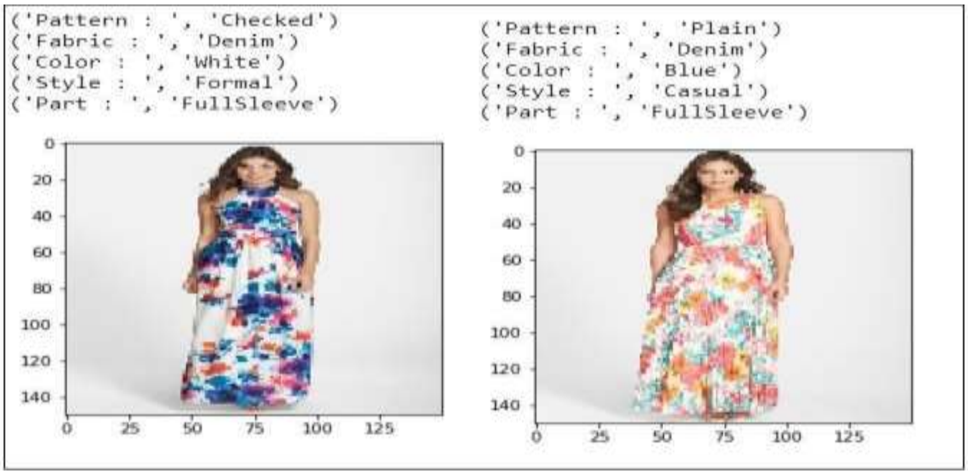

Figure 6. Input images with predicted values

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.

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.

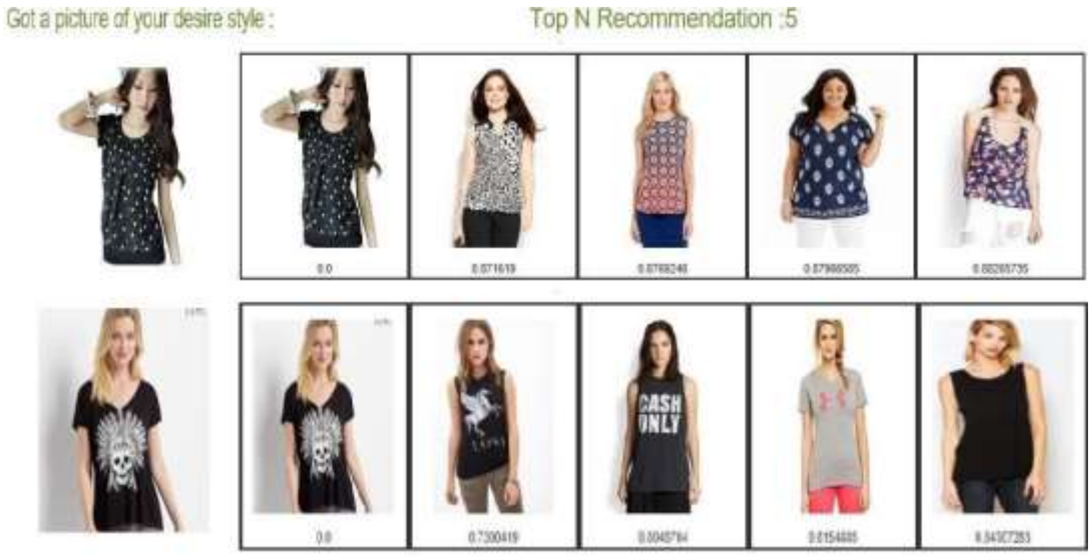

Figure 7. Pattern recommendation

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.

Figure 8. Outside recommendation dataset

IMPLEMENTATION AND OUTPUT SCREENSHOTS

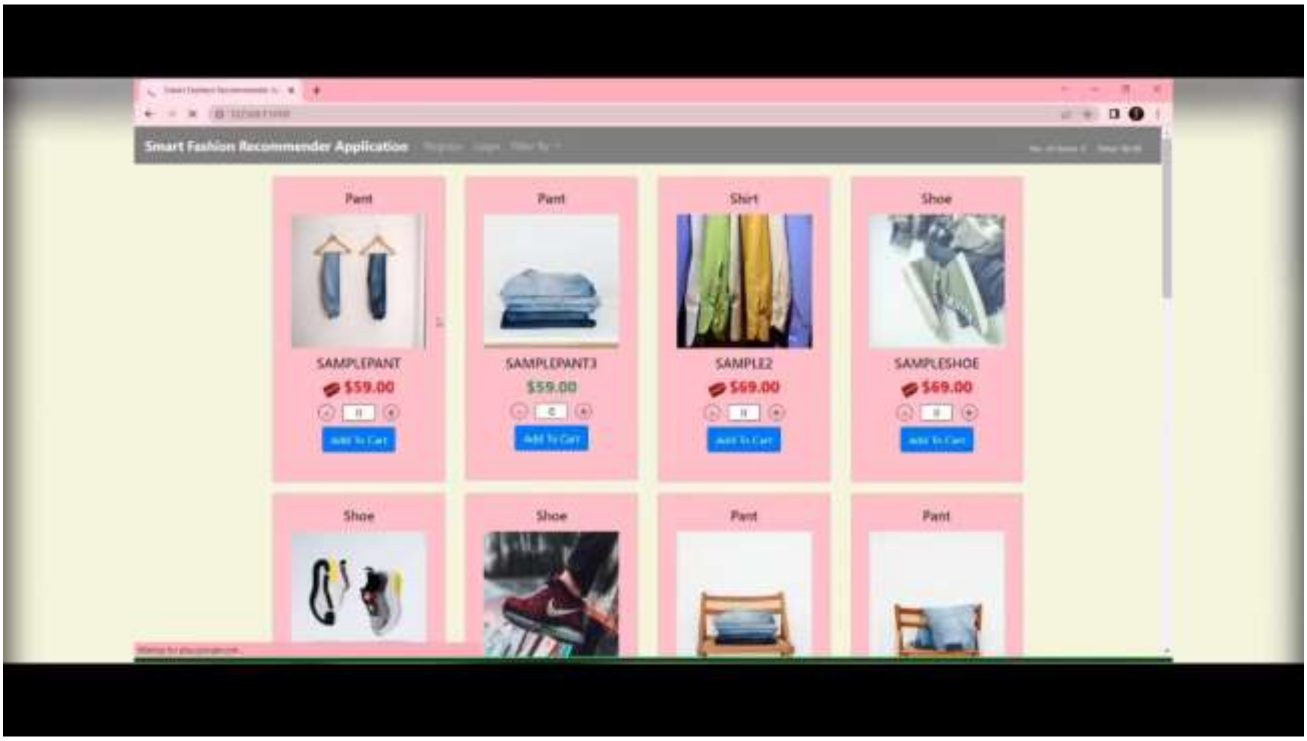

Figure 9. Home Page

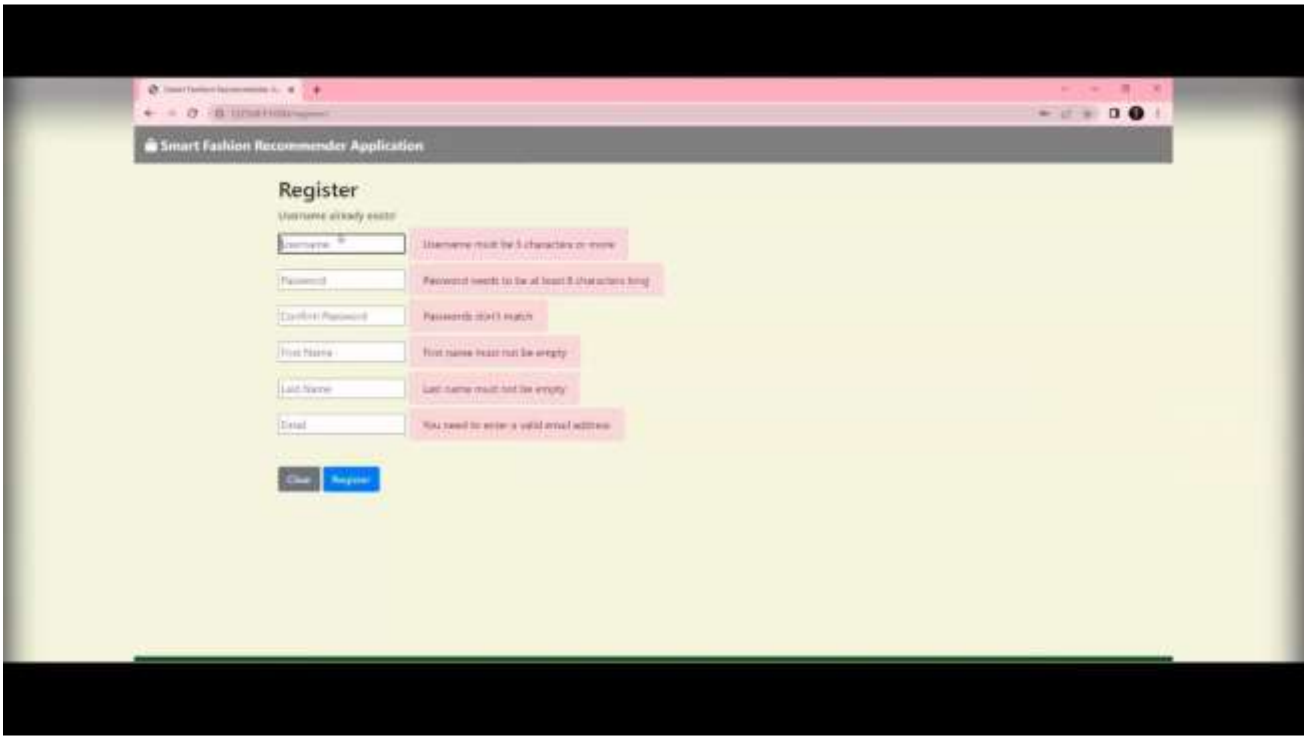

Figure 10. Registration page

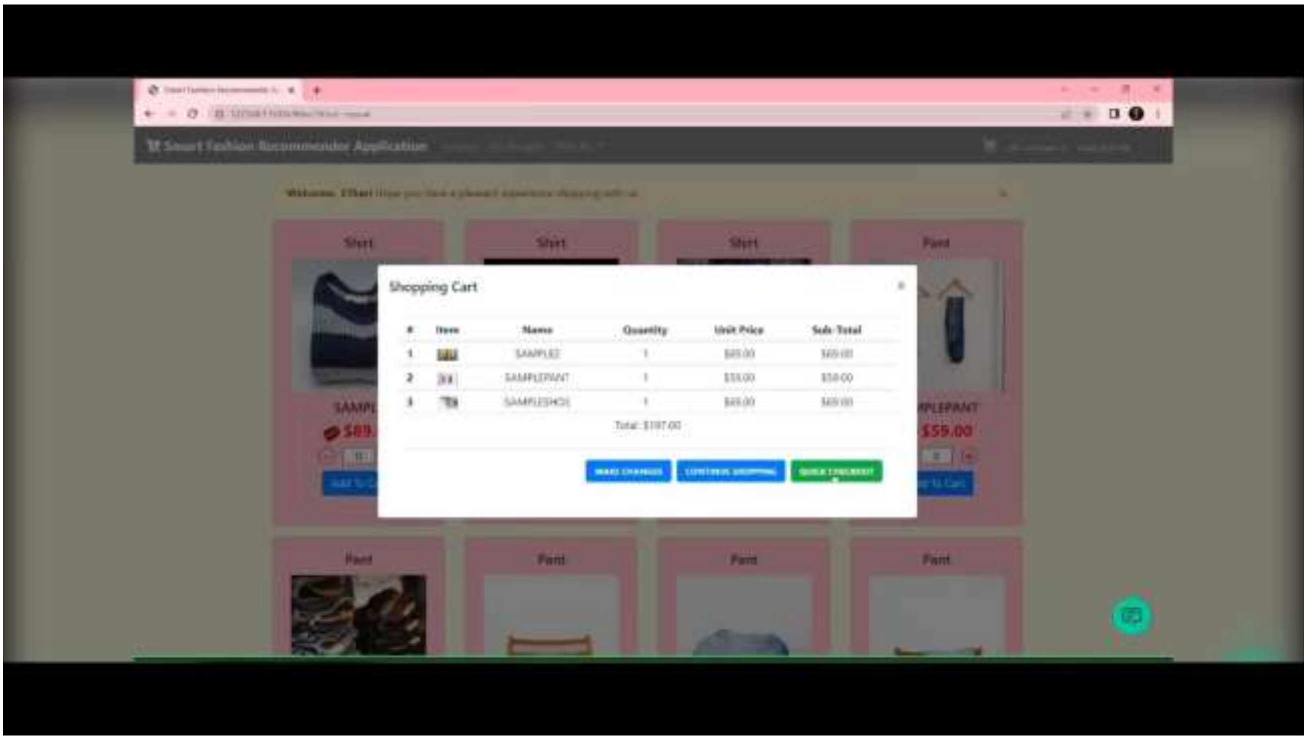

Figure 11. Shopping cart window

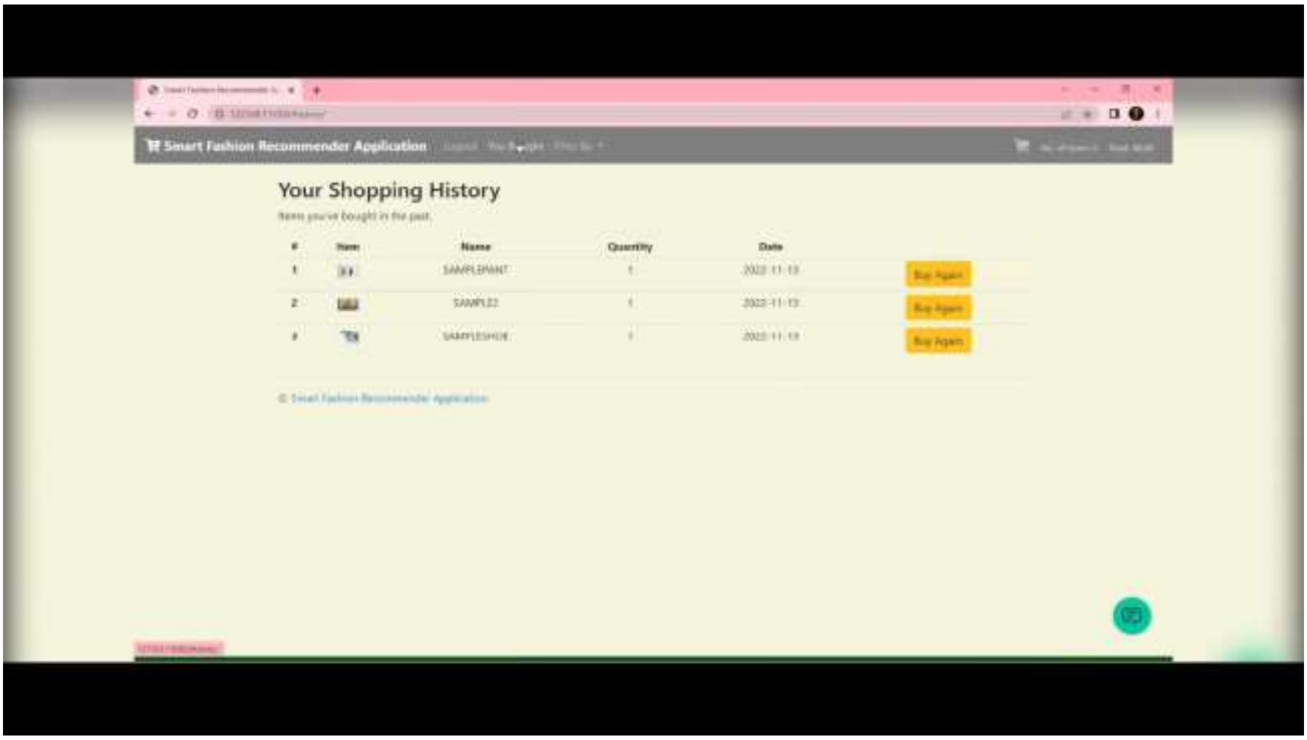

Figure 12. Shopping cart Page

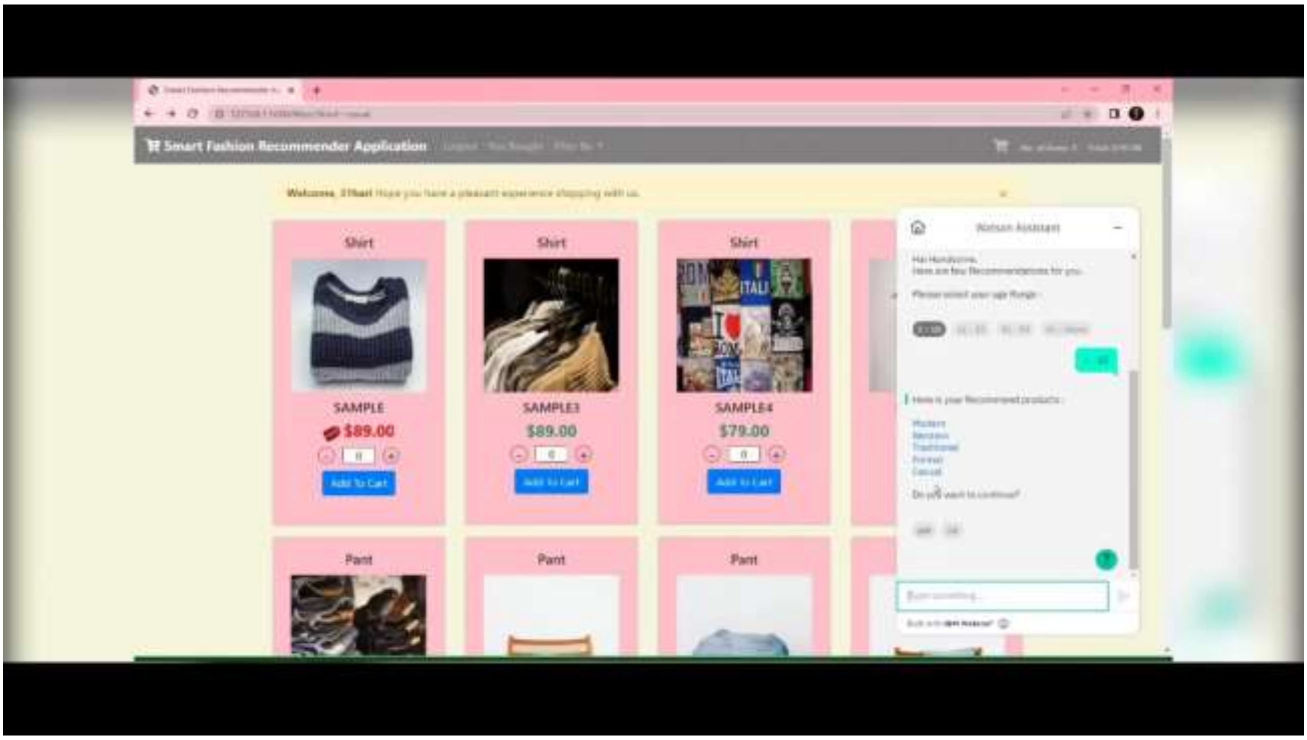

Figure 13. IBM CHATBOT

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.