SCAD COLLEGE OF ENGINEERING AND TECHNOLOGY
ANNA UNIVERSITY NOVEMBER 2022

SMART FASHION RECOMMENDER APPLICATION

A PROJECT REPORT
Submitted by

R.ALAGU SENTHUR (952819104003)
A.ALDO JENUS (952819104004)
A.GOLDWIN AUGUSTIN (952819104013)
S.MOHAMED RIYAS (952819104028)

In partial fulfllment for the award of degree of Bachelor of Engineering (B.E) in Computer science Engineering

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our Faculty Mentor and Industry Mentor for their support and guidance in completing our project on the Smart Fashion Recommender Application.

We would like to extend our gratitude to the IBM for Nalaiya Thiran project for providing us with all the facility that was required.

It was a great learning experience.We would like to take this opportunity to express our gratitude.

TEAM MEMBERS:

R.ALAGU SENTHUR (952819104003)

A.ALDO JENUS (952819104004)

A.GOLDWIN AUGUSTIN (952819104013)

S.MOHAMED RIYAS (952819104028)

TITLE

ABSTRACT

In recent years, the huge amount of informaton 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 utlizaton of informaton, also known as the informaton overload problem. Traditonally, keywords are used to retrieve images, but such methods require a lot of annotatons on the image data, which will lead to serious problems such as inconsistent, inaccurate, and incomplete descriptons, and a huge amount of work. To solve this problem, Content Based Informaton Retrieval (CBIR) has gradually become a research hotspot. CBIR retrieves picture objects based entrely 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 specifc shapes, colours, and textures. These visual characteristcs of the image are used as input conditons 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 atributes from images with clothes to learn the user's clothing style and preferences. These atributes are provided to the correspondence model to retrieve the contguous related images for recommendaton. Based on data-driven, this thesis uses convolutonal neural network as a visual extractor of image objects. This experimental

model shows and achieves beter results than the ones of the previous schemes.

Recommendation systems are the techniques that are used to predict the rating one individual will give to an item or social entity. The items can include books, movies, restaurants and things on which individuals have different preferences. These preferences are being predicted using two approaches frst content-based approach which involves characteristics of an item and second collaborative fltering approaches which considers user's past behaviour to evaluate its choices. This thesis proposes a fashion recommendation system which will recommend clothing images supported the style sort of the provided clothing images. In this work, we focus on the images of upper body as well as the lower body clothing and with human model in the images. We have created our own datasets through web scrapping of different ecommerce websites. In this paper we have come up with an idea to build a content-based recommendation system using ResNet-50 convolutional neural network.

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

Introduction

During the last years, online shopping has been growing. In 2013, the total turnover for ecommerce in Europe expanded with 17% in contrast to the 12 months before and huge organizatons 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. 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. 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 informaton uncover that women are more likely to buy on-line, with 71% of ladies doing this, contrasted with 52% of men. 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. 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 informatonal collecton associatons with PC vision, looking into the issue as indicated through two uncommon perspectves, the frst being text-based and the second one being visual-based. From the outset, the developments have been made only through informaton annotatons that have been saved in a database to work the retrieval step, however, when the dimension of the image collectons started to amplify the efort required to label them used to be as soon as unsustainable, to solve this issue, during the 1990s, content-based photograph retrieval was proposed. Startng now many searched for lines have seemed the use of one or the diferent isolated or combining them. Recommendaton systems make recommendatons based on the informaton they are provided with and in the manner in which they are programmed. Going into details, most of the evaluaton applied is independent coming up with a brand-new recommendaton algorithm, system, or model. However, diferent researchers use already existng work as researchers use an already existng 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. Existng research and fashions have given us some inspiratons of how to design fashion recommendaton 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 suggestons based on provided contents.

The contributon 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.
- These atributes served to a similar model to retrieve similar images as recommendatons.
- Combined with more common content-based recommendaton systems, our model can help to extend robustness and performance.

Literature survey

Myntra-Matching Clothes Recommendation:

On selecting a particular item to buy, Myntra automatically suggests a full set of clothes that are matching to the selected item. For example, on selecting a particular t-shirt, the system automatically generates a combination of watches, shoes, pants, etc. that are matching to the selected t-shirt. This system does not take into consideration private qualities of customers like skin color and existing clothes. It will only suggest clothes that already exist in its database.

Your Closet:

This is a mobile application that organizes the closet. The user interface is shown in. The application asks customer to input their clothes. It then matches each cloth with other clothes. For example, if there are 4 shirts and 4 pants, the application matches each shirt with each pant and thus provides 16 possibilities. The application does not make matches of clothes depending upon patterns, color and texture of clothes. It also does not have a recommendation system.

Your Closet App Magic Closet:

This system aims to retrieve clothes from online stores that are matching to the input clothes. These clothes must be fit to a particular occasion. In this system, the user takes a photo of them specifying if they want to use the top or bottom clothes along with the occasion they want to use it for. The system will search for clothing that matches the user query and satisfies the criterion of wearing aesthetically and wearing properly.

Which Clothes to wear confidently?:

The basic problem the system addresses is: From the two given images corresponding to a pair of clothes, we have to determine if the pair of clothes matches or not. While there may be several aesthetics espoused by different individuals, it takes a simplistic approach in this problem. An example of shirts and ties is used. Various machine learning methods are used to classify if the clothes are matching or not such as Ridge Regression, Standard Neural Network and Siamese Neural Network.

Personalized Clothing Recommendation Based on Knowledge Graph:

This system attempts to exploit the knowledge graph for providing clothing recommendations to the user keeping the user context in mind. The recommendation is done by calculating the similarity in the clothing ontology similar to users collection. Skin and Clothes matching seeded by Color System Selection: The main aim of the system is to suggest clothes to user based on skin color. The paper first finds out which color scheme is best suited to represent skin colors and then tries to find a way to recommend if clothes and skin color match. An automated system to determine the highest levels of color suitability between skin and clothing was made.

Discerning Advisor:

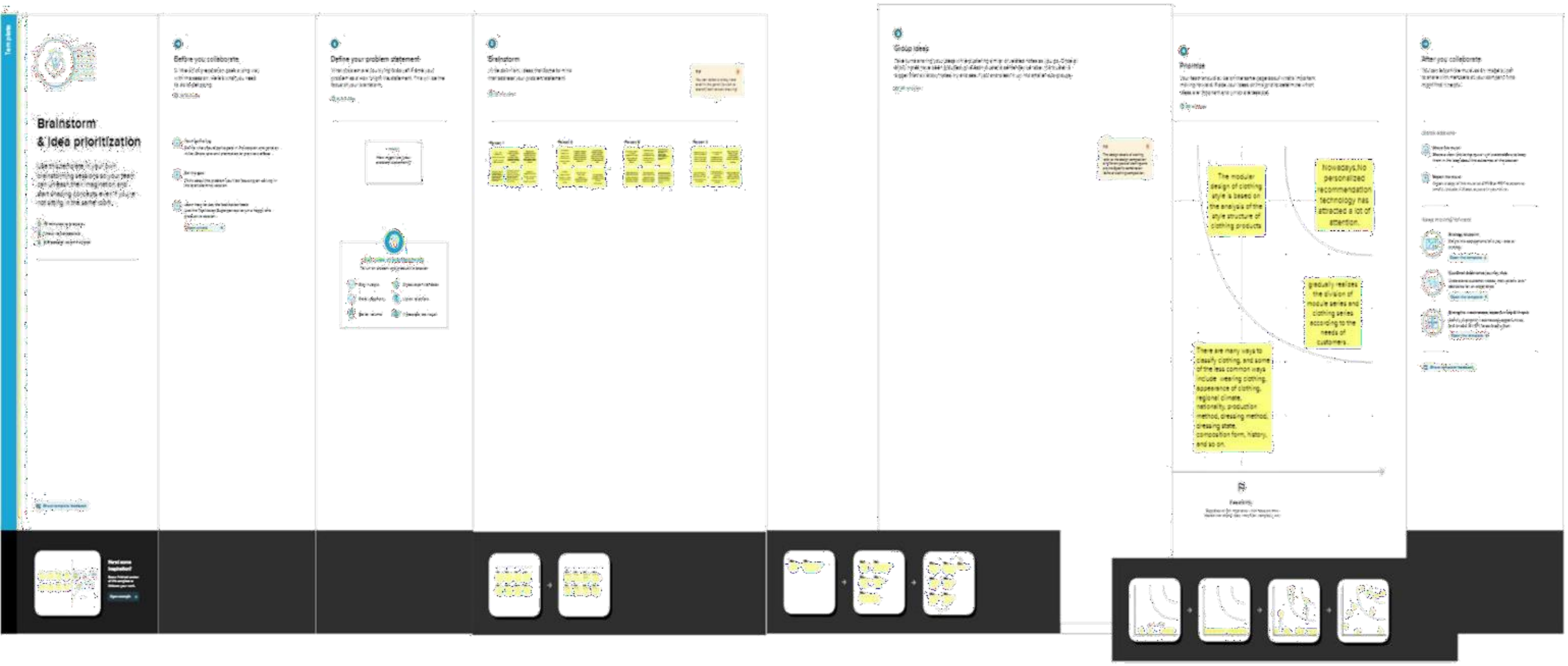
The system tries to recommend clothes based on skin olor of the customer. Using a neural network, frst the skin color is detected. Fuzzy logic is used to map a skin color to the skin color of a fashion model, and clothes suited to that model are recommended. Garment Detectives: The garment detection is to detect the presence of clothes in images and somewhat locate their extents, where the localization can be defned from coarse (image) level to fne (pixel) level. A unifed system is proposed for detecting and recognizing clothes in customer photos.

Identifying Corners of Clothes by Image Processing:

This system aims to fnd the edges of the clothes for clothes manipulation. This system achieves this by fnding pixels that represent the clothes. This system frst accepts user image and then performs several image processing operations to improve the efciency of edge detection. It then uses certain criterion to decide whether a pixel represents an image or not.

Real-time Clothing Recognition from Surveillance Videos:

It is an analysis system of contents of video which is capable of tagging various clothes of different persons is created. First, face detection and tracking is performed and each frame is aligned. The system then proceeds to clothing segmentation using a variant of region growing method. Through this, clothes are detected. The system then proceeds to clothing recognition and indicates the type of clothing – skirt, t-shirt, etc.



Template

Brainstorm & idea prioritization

Before you collaborate

Define your problem statement

Brainstorm

Group ideas

Prioritize

The modular design of clothing style is based on the analysis of the style structure of clothing products

Nowadays,No personalized recommendation technology has attracted a lot of attention.

gradually realizes the division of module series and clothing series according to the needs of customers.

There are many ways to classify clothing, and some of the less common ways include wearing clothing, appearance of clothing, regional climate, nationality, production method, dressing method, dressing state, composition form, history, and so on.

Feasibility

After you collaborate

Proposed system architecture

The system architecture defnes the hardware, sofware 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 vertcally. 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 vertcal way.

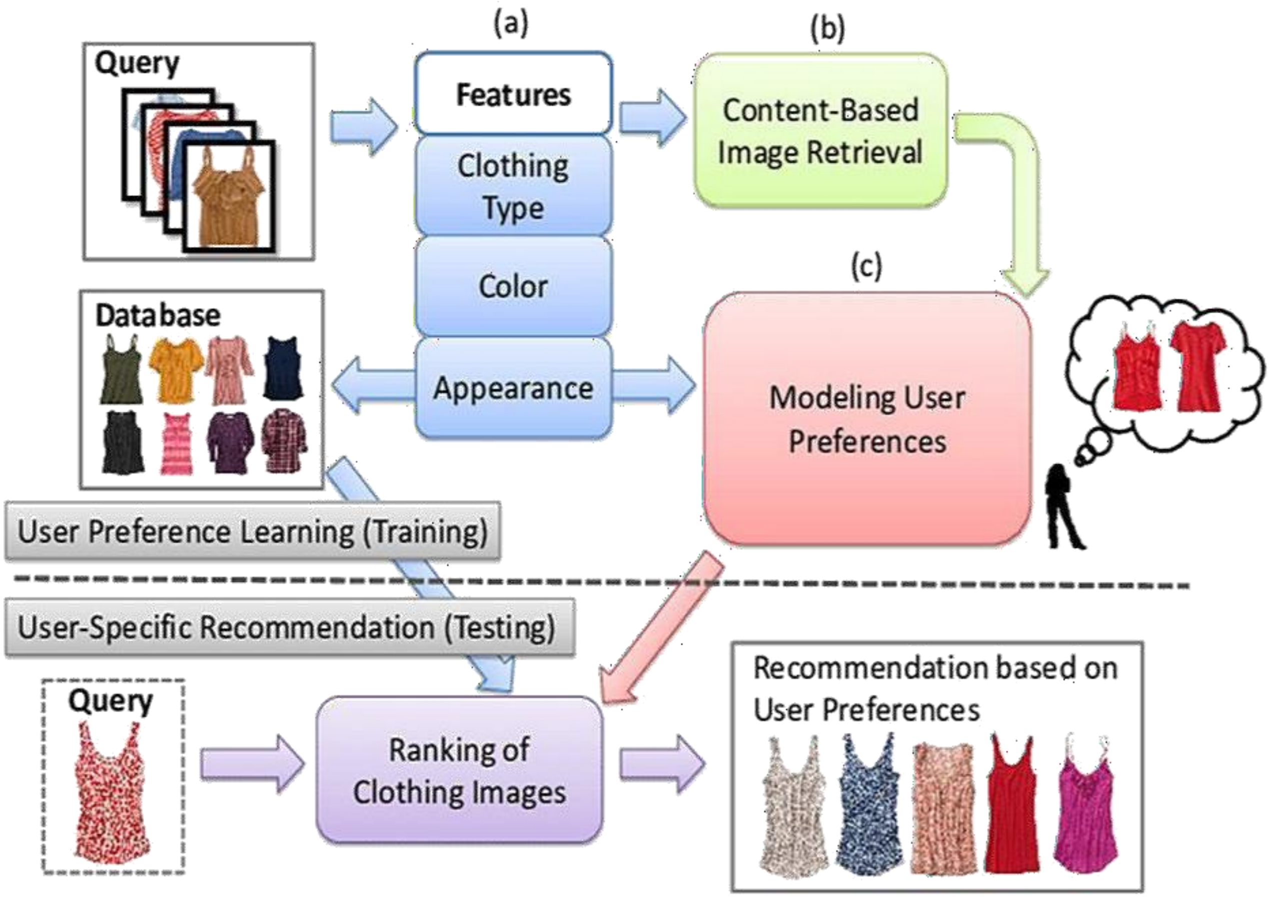

(a)
(b)
Query
Features
Clothing Type
Color
Appearance
Content-Based Image Retrieval
(c)
Database
Modeling User Preferences
User Preference Learning (Training)
User-Specific Recommendation (Testing)
Query
Ranking of Clothing Images
Recommendation based on User Preferences

Figure 1. **System architecture**.

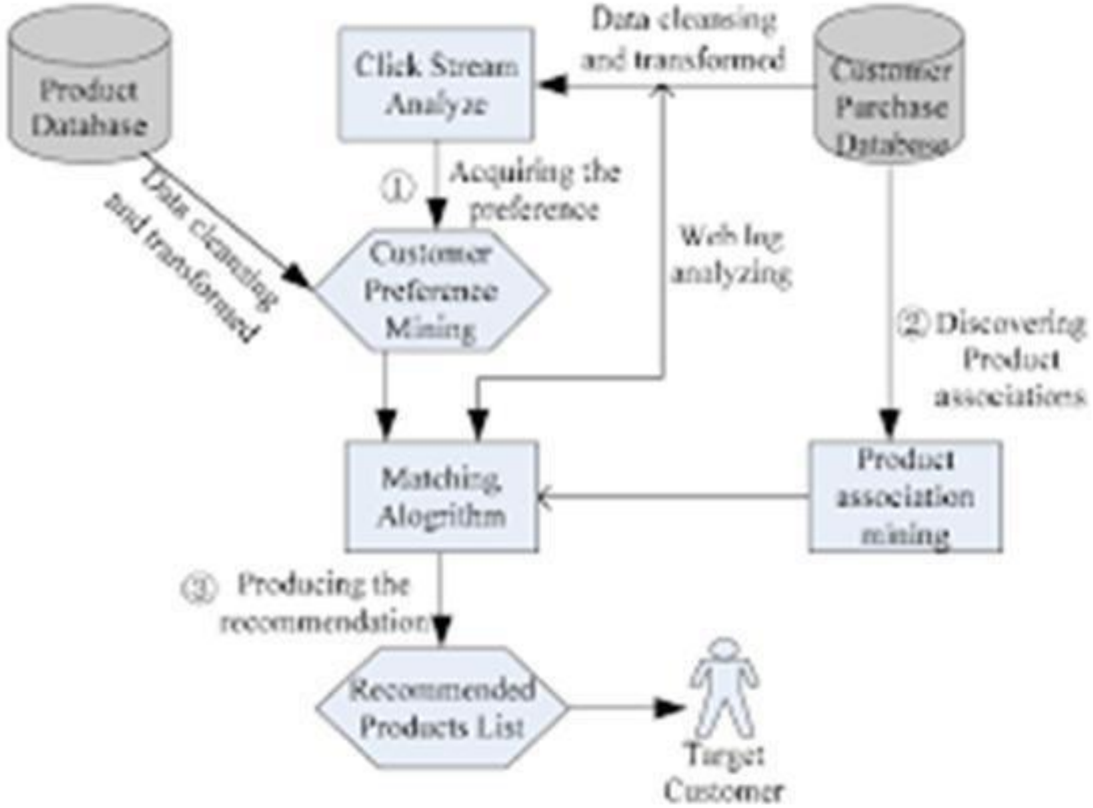
The system comprises of the Client tre, which is the front end or View mode, middle ter which is the system controller and the backend tre which is the model. The client side is where the users/customers log in in the system, browse for the system interface, provide input query image to the system, and get recommendaton according to the input query.

The middle true is responsible for communicaton 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 internet works to provide access to the site with a strong security check, provided by both frewall and password protecton policy.

Any unauthorized access is detected and prevented by the firewall.

a. The vertical classification system model

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. Receives Recommendations User Web Server Sends response to user Database Stores User input data Stores dataset features.


Product Database
Click Stream Analyze
Data cleansing and transformed
Customer Purchase Database
① Acquiring the preference
Data cleansing and transformed
Customer Preference Mining
Web log analyzing
② Discovering Product associations
Matching Alogrithm
Product association mining
③ Producing the recommendation
Recommended Products List
Target Customer

Data Recommender Makes Recommendations Recommender Algorithm Determines the Similarity between cloths Figure 2. Vertical architecture of the system. 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.

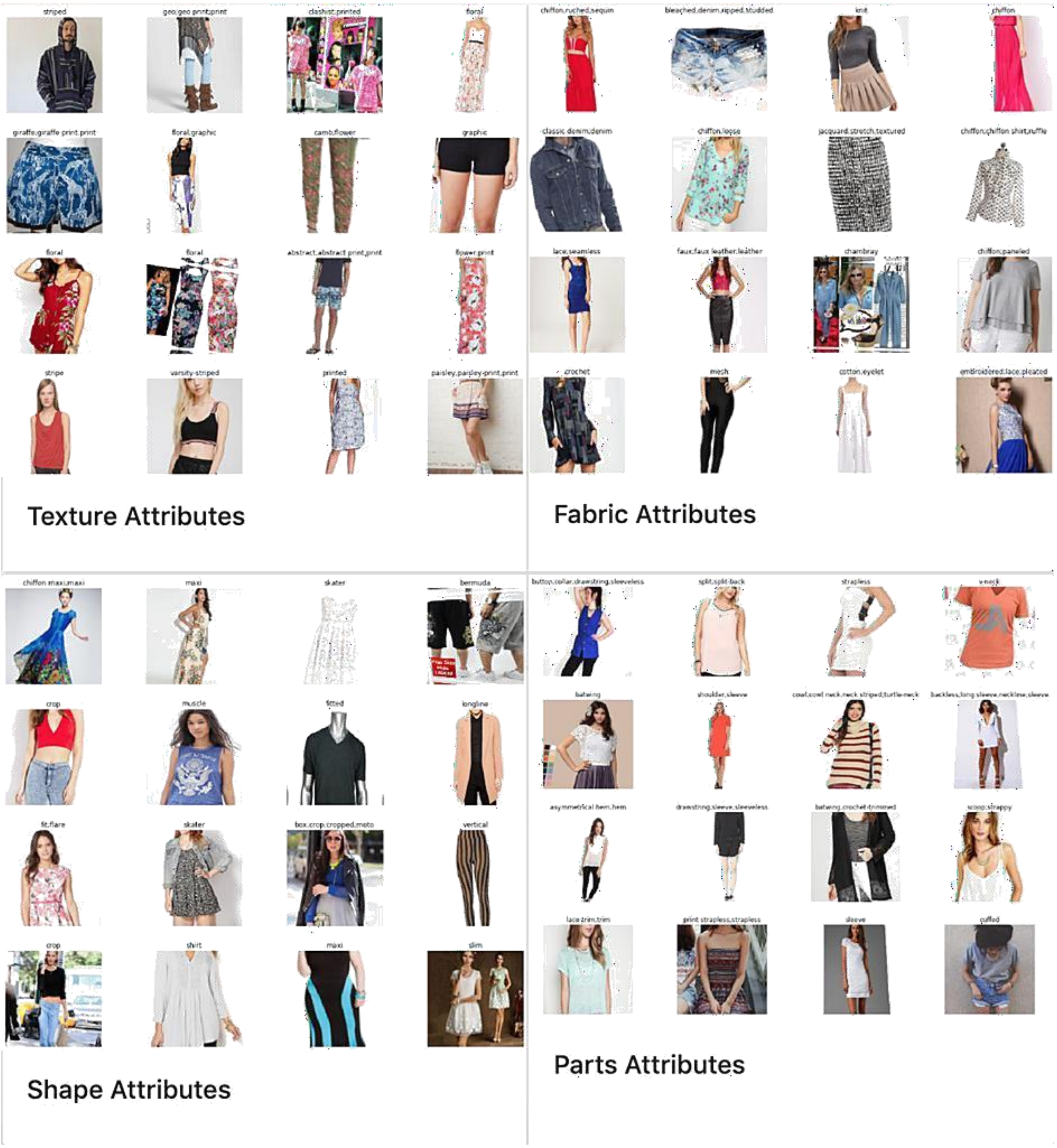
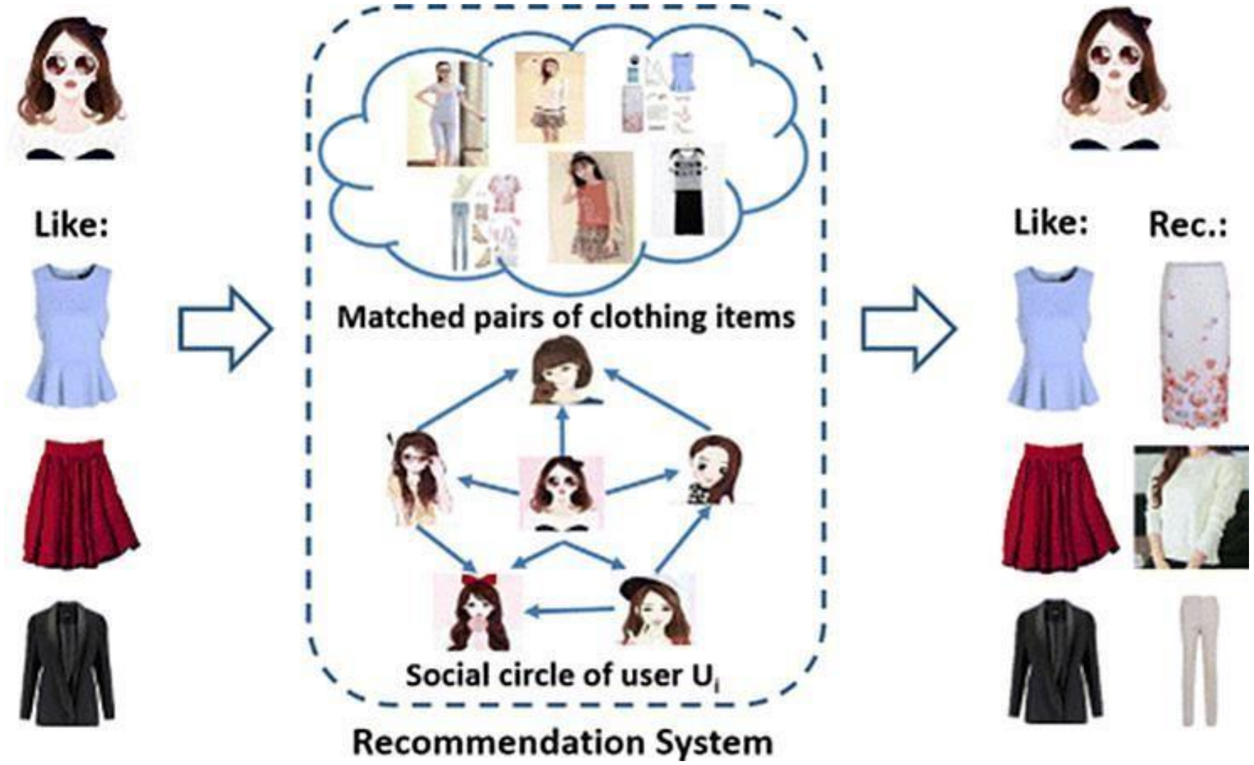


Figure 3. Fashion dataset One subset, known as Atribute Predicton, can be used for apparel category and atribute predicton. From almost 290,000 photos of 50 apparel categories and 1,000 apparel atributes, we randomly picked 18k images from diferent categories and then we classifed them for training and testng. The distributon of labels is presented in Figure

c. Design of deep learning module

There are many classifcaton algorithms or classifers in use today. The most notably and the most implemented classifers and feature extractor are implemented to solve a problem of cloth / fashion recommendaton Design process. (1)are weight vectors, are fully connected output layers that actually perform classifcaton and are the CNN without the last layer. They are used as a feature extractors.



The core network of our model as shown in Figure 4. who presented a convolutonal neural network in the paper “Very Deep Convolutonal Networks for Large-Scale Image Recogniton”, at the University of Oxford.”. Then model is checked for top-5 accuracy on ImageNet.

Results and Evaluation

This section focuses on evaluating our system and deciding the stage which it is able to fulfil 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. 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

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 profle. This is brought out by taking one or more noteworthy pictures of the client's ideal atre things as they were entered and by making their style vector. These vectors are then blended to shape the framework of the individual style profle. The Figure 5 shown Patern recommendaton with similarity score.

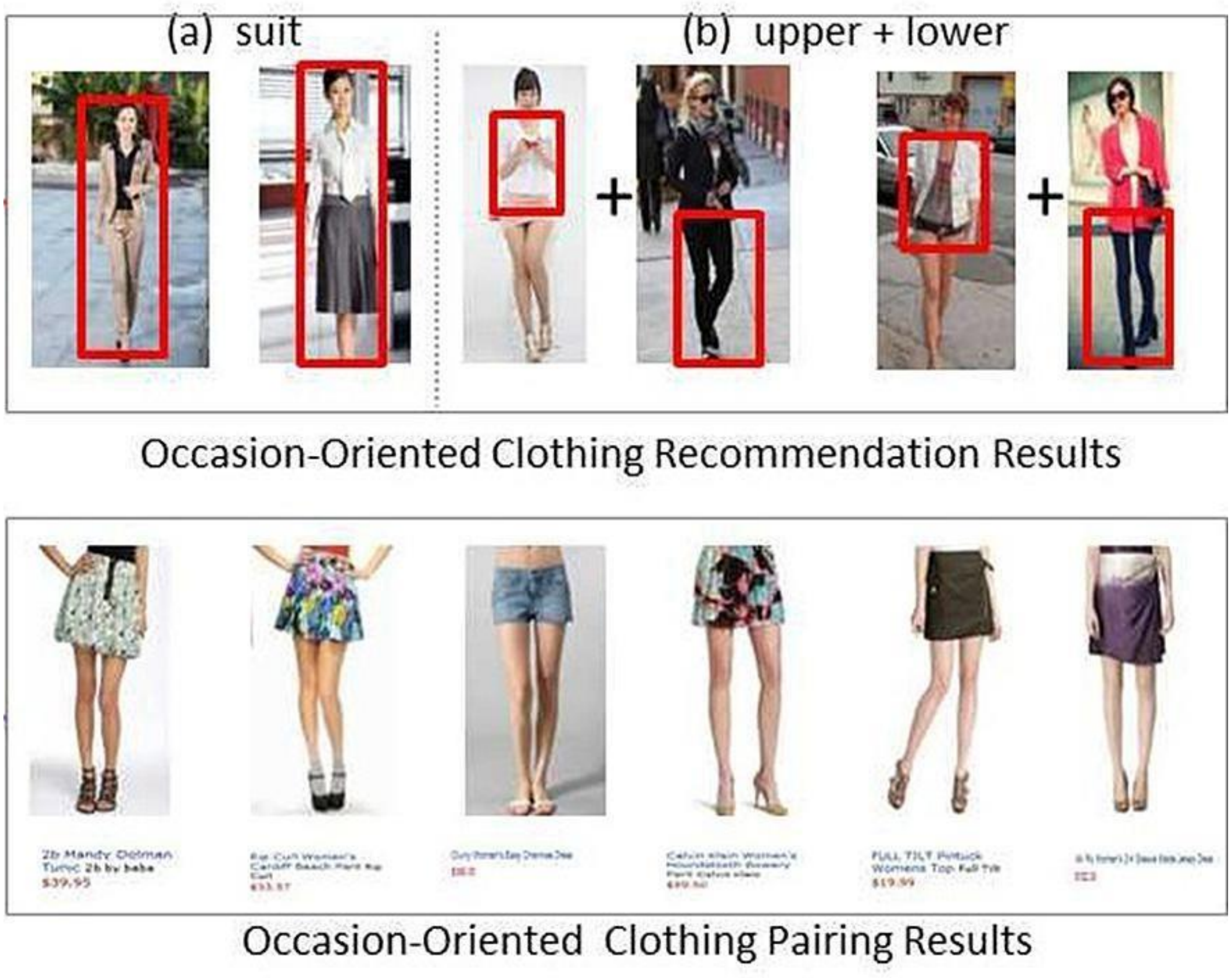

(a) suit
(b) upper + lower
+
+
Occasion-Oriented Clothing Recommendation Results
2b Mandy Dolman Tunic 2b by bebe
$39.95
Calvin Klein Women's Houndstooth Bowery
FULL TILT Pintuck Womens Top Full Tilt
$19.99
Occasion-Oriented Clothing Pairing Results

Figure 5. Patern recommendaton with similarity score

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

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 platorm through which a user can visit the system and provide his/her choices regarding the fashion images for best recommendaton.

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 recommendatons. Afer making the fashion vector, some predictons are made, as illustrated The system is responsible for making recommendatons to users based on their user data. The user data compiled in the dataset is fltered by the recommender system through the recommender algorithm.

Step1: Def similarity (feature_data, inp_feature_data); Num_samp=inp_feature_data.size

Step2: print (num_samp) Sim_score = () For i in range (len (feature_data)); score = 0 show_sample (data_images[i]) print(feature_data[i]) Romanian Journal of Informaton Technology and Automatc Control, Vol. 31, No. 4, 123-136, 2021 131 http://www.rria.ici.ro **Step3:** Score_m inp_feature_data-feature_data[i] print (score_m)

Step4:Score=num_samp-np.count_nonzero(score_m) Sim_score[i]=score print(score)

```
Step 5: Return sim_score


Step 1: Similarites=similarity(feature_data,inp_feature_data)
step.2.items(),
                 key=operator.itemegter(1),
                 reverse=(true)
                 Num_reco=30
data=feature_data.size
                 For I in range(num_reco)
                 Ind = sorted_similarites[i][0]
           print (sorted_similarites)

                 Print ("score:", sorted_similartes[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, colour, 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 colour 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.

DESCRIBING CLOTHES AND STYLE

CLASSY

Your style is elegant and, with clean, simple lines, softly tailored not usually mix and match. Never trendy, faddish or severe, Textures such as silk, soft woolen fabrics, cotton are among your favourites.

FUNKY ROCK

This look is edgy, a fashion style expressed by bold hair styles and colors, accessories, black leather jackets and shirts with funky designs.

ETNIC

The easiest way to wear the look is a print. If prints aren't your thing, then you can still indulge in the ethnic /tribal fashion trend by way of accessories.

GIRLY GLAM

Your outfits hug Your body tightly and are full of pink, loud and sparkly accessories, high stilettos and platforms. It is a style that exposes some of your body and brings out a flirty, feminine and sexy look.

BOHO CHIC

It a vintage inspired style with earthy tones, loose fitting clothes, and combining certain looks to appeal to an eclectic vibe. This trend is a mix of luxurious textures and contrasting fabrics, big purses, casual, long necklaces...

TRENDY

You love the big city and adore your uniform of all black with stiletto boots, pumps, or sandals. You love to dress up, be in all the latest trends. You know which pieces make the whole look appear expensive.

PREPPY/GIRLY

Polo shirts, or Oxford shirts, dresses and T shirts all either in pastel or bright colors. jeans, khakis, slacks, or Bermuda shorts. Oxford shoes, tennis shoes, ballet flats make the look.

CASUAL

A cute, sweat suit is your go-to outfit .If an evening activity is on the agenda, a quick change to slacks or jeans and a sweater or nice T-shirt does the trick. Your color navy, gray, and black

PUNK

A lot of punk clothing has a DIY (do it yourself) look to it. Flight jackets, skinny jeans, leather, studs, are the things to wear together with classic accessories like, arm warmers, bullet belts, and wristbands with pyramid studs, stars or spikes.

MOD

Dresses and skirts are A-line and either mid-thigh length or just below the knee. Go for crazy patterns or stripes with lines that pop. Typical colors are white and black, but you can incorporate others. Mod shoes are chunky knee- or ankle-high boots

SKATER

Their look is more put together and creative then you'd think. Strategic layers (long and short sleeve Tees) and smart yet accessories (necklaces, etc.) are the basis of this look.

HIPSTER

Wear skinny jeans, basically, the tighter, the better. Wear glasses and dress vintage. Hipster shoes include cowboy boots, Converse, and a range of flats. Wearing things that don't match together is also very hipster.

HOBO

Hobo's clothes must be either painfully small or comically large. Hobo Shoes can cover a wide range of styles, as long as they are beaten up and battered, fingerless gloves and hats give you the look too.

GOTH GLAM

Goth glam fashion is now a classy version of gothic. It still retains dark and gloomy colors like black, navy and burgundy.

ROMANTIC

You like to wear soft colors and fabrics like chiffon and silk that drape graciously on your body. You also go for ruffle or skirt and dresses with floral prints, lace, crocheted tops, cameos and antique jewelry.

EMO

Tight clothing is still your thing, though much of it has taken a more modern look now. Stripes are popular as stars, hearts, skulls, nautical stars, and sparrows. Hair still black.

As shown in Figure, 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 colours, 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.

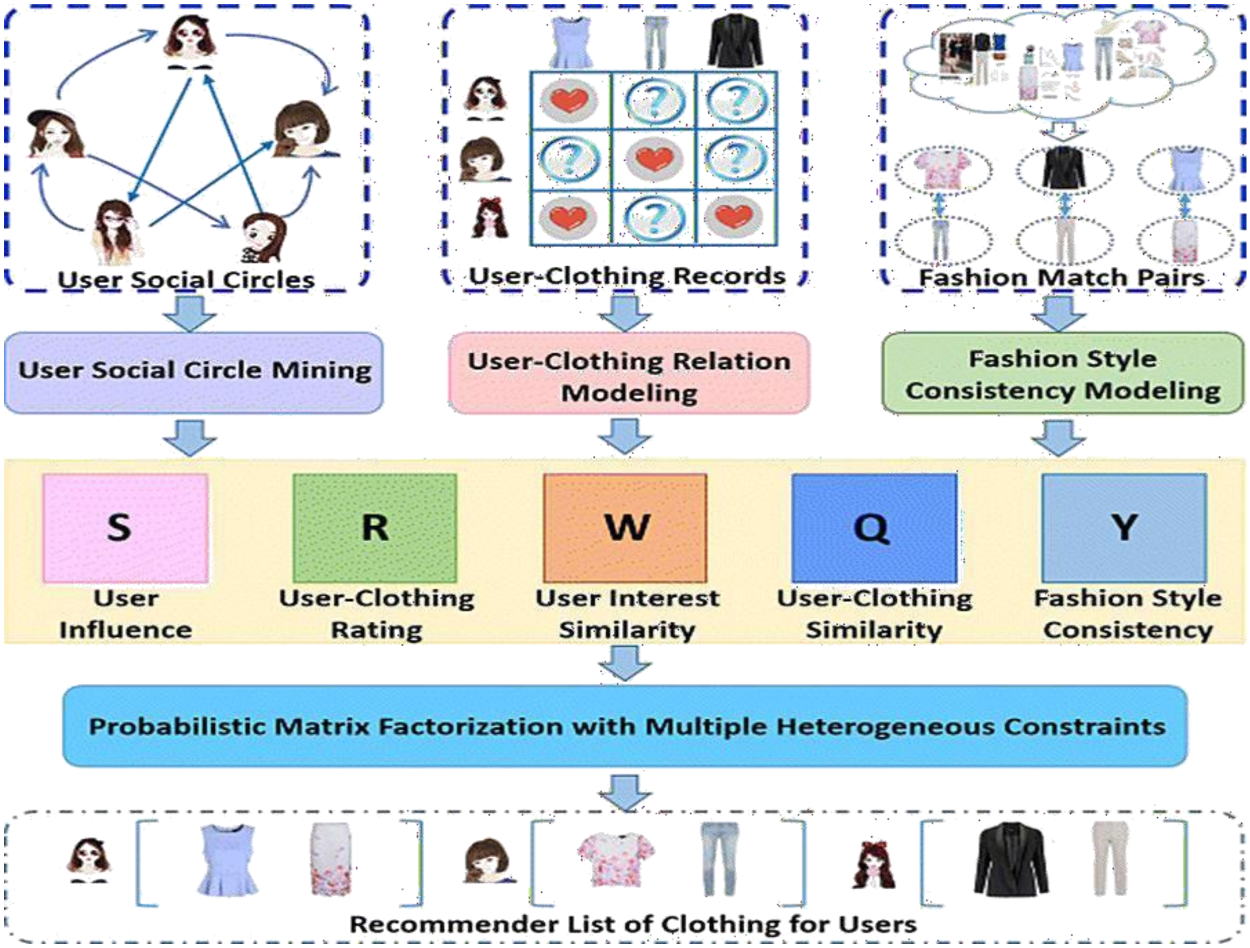

User Social Circles
User-Clothing Records
Fashion Match Pairs
User Social Circle Mining
User-Clothing Relation Modeling
Fashion Style Consistency Modeling
S
User Influence
R
User-Clothing Rating
W
User Interest Similarity
Q
User-Clothing Similarity
Y
Fashion Style Consistency
Probabilistic Matrix Factorization with Multiple Heterogeneous Constraints
Recommender List of Clothing for Users

Figure: Outside recommendaton dataset

System result and accuracy:

Finally, this subsecton evaluates the system and shows the testng results and the accuracy of our model. Afer adding the model on top of the convolutonal 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.

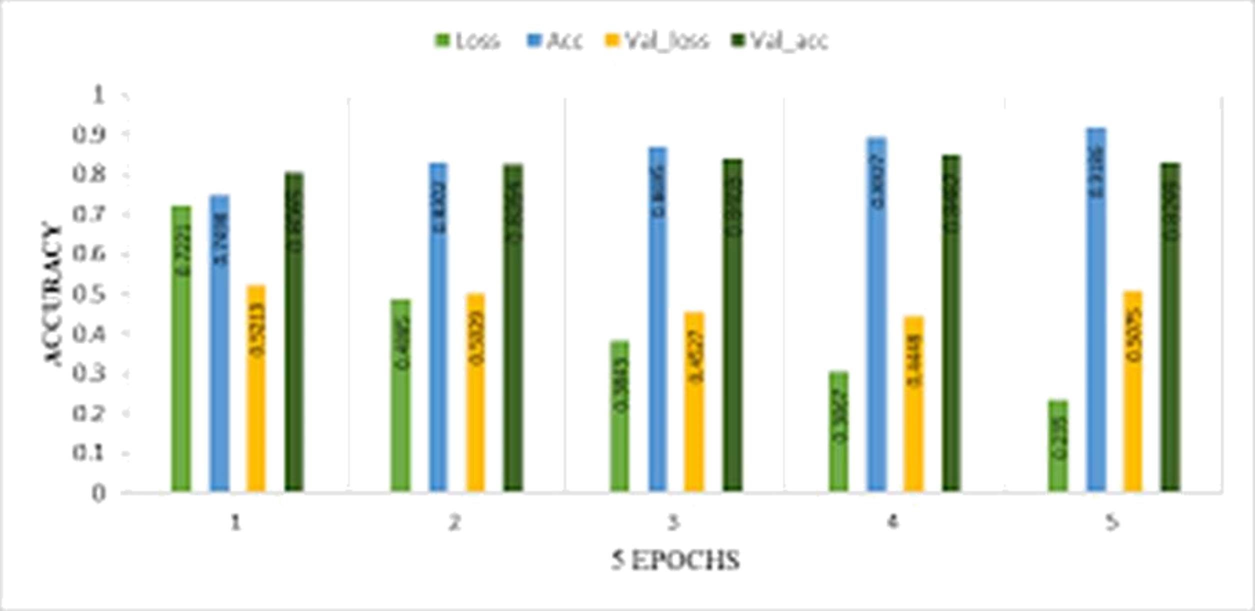


Figure 9. Model accuracy afer freezing the layers for 5 epochs

Afer calculatng the mean accuracy for 5 epochs, the obtained results are as follows:

Validaton: accuracy = 0.836000; loss = 0.489109

This part of the sentence “Afer calculatng the mean accuracy for 5 epochs” is mentoned also below, afer Figure 10, and these values mentoned for accuracy and loss (0.836000 and 0.489109) are not illustrated in Figure 9, but in Figure.

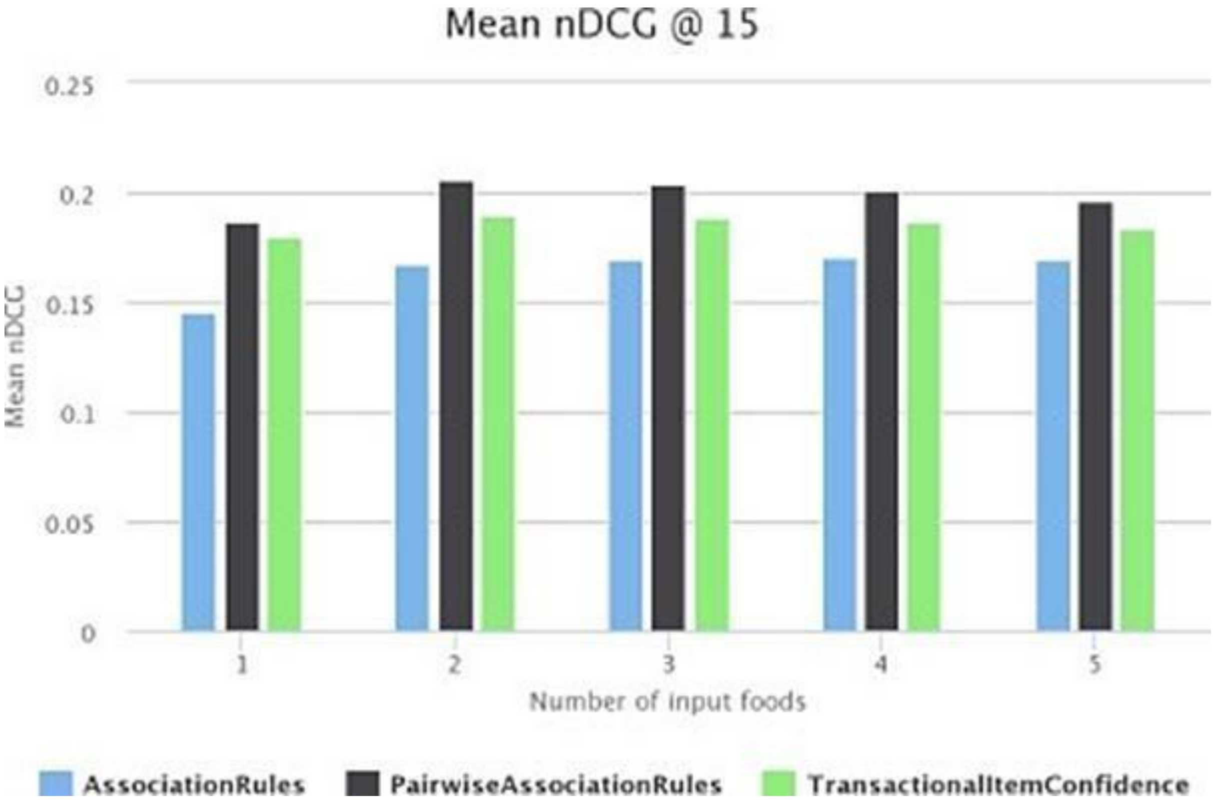


Figure 10. Model accuracy for 5 epochs

Afer calculatng the mean accuracy for 5 epochs, the fnal result are as follows: Validaton: accuracy = 0.864750; loss = 0.516400 These values mentoned 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 notced that our model gives a beter accuracy when compared to Alex Net, as shown in Figure 11.

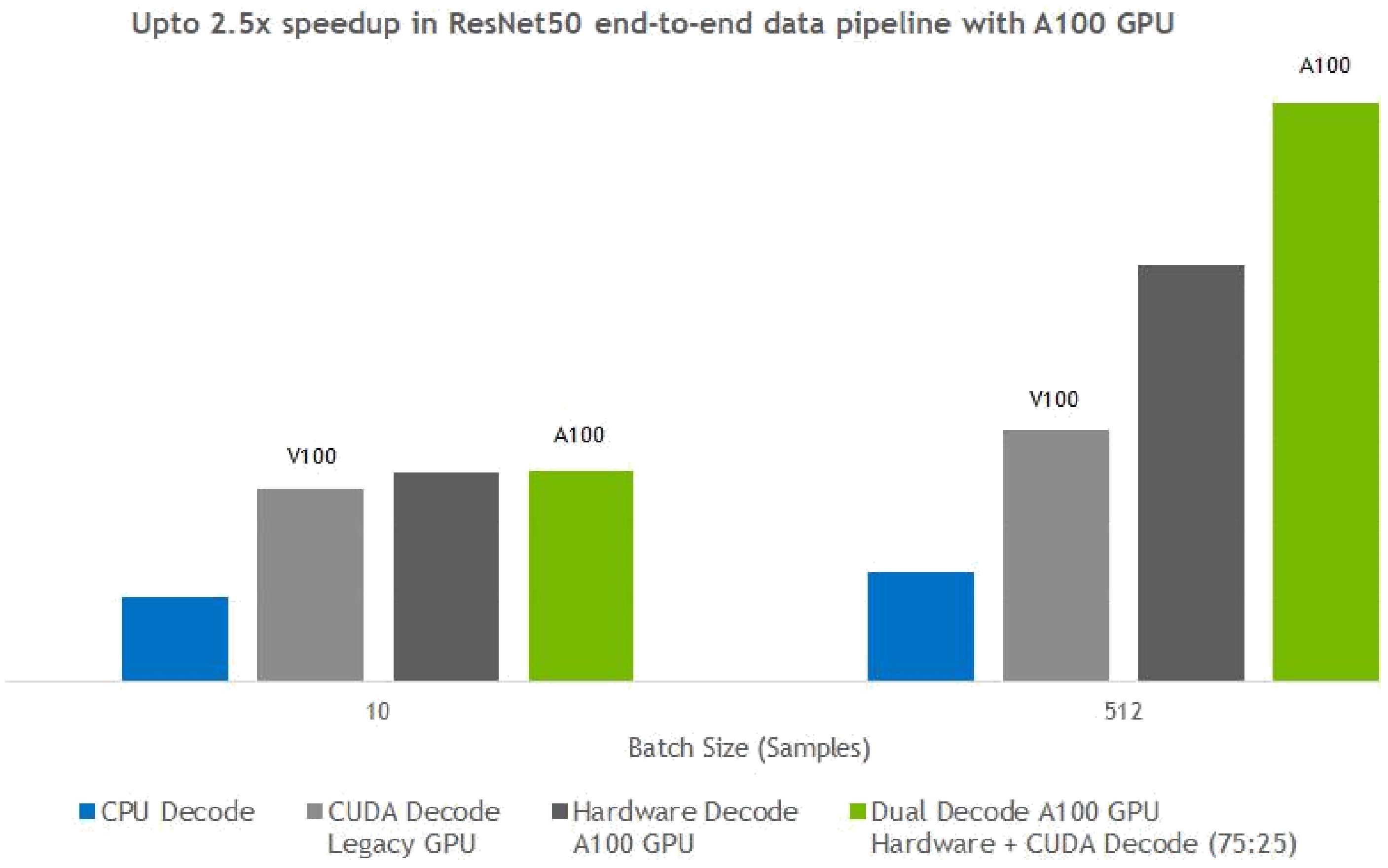


Figure 11. Accuracy and loss

Recommender systems are stll 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 soluton to the overload of web data, an important problem afectng 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.

Source code:

adminlogin.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Log in</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');
        @import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
body{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-color: #23242a;

}
.box{
    position: relative;
    width: 380px;
    height: 400px;
    background-color: #1c1c1c;
    border-radius: 8px;
    overflow: hidden;
}
.box::before{
    content: '';
    position: absolute;
    top: -50%;
    left: -50%;
    width: 380px;
    height: 420px;
    background: linear-gradient(0deg, transparent,#45ff55,#45ff55);
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
}
.box::after{
    content: '';
    position: absolute;
    top: -50%;
    left: -50%;
    width: 380px;
    height: 420px;
    background: linear-gradient(0deg, transparent,#45ff55,#45ff55);
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
    animation-delay: -3s;
}
@keyframes animate {
    0%{
        transform: rotate(0deg);
    }
    100%{
        transform: rotate(350deg);
    }
}

/* Form */
.form{
    position: absolute;
    inset: 2px;
    border-radius: 8px;
    background: #28292d;
    z-index: 10;
    padding: 40px 40px;
    display: flex;
    flex-direction: column;
}
.form h2{
    color: #45ff55;
    font-weight: 500;
    text-align: center;
    letter-spacing: 0.1em;
}
.inputBox{
    position: relative;
    width: 100%;
    margin-top: 35px;
}

.inputBox input{
    position: relative;
    width: 100%;
    padding: 10px 10px 10px;
    background: transparent;
    border: none;
    outline: none;
    color: #23242a;
    font-size: 1em;
    letter-spacing: 0.05em;
    z-index: 10;
}
.inputBox span{
    position: absolute;
    left: 0;
    padding: 20px 0px 10px;
    font-size: 1em;
    color: white;
    pointer-events: none;
    letter-spacing: 0.05em;
    transition: 0.5s;
```

```
}
.inputBox input:valid ~ span,
.inputBox input:focus ~ span{
    color: #45ff55;
    transform: translateX(0px) translateY(-34px);
    font-size: 0.75em;
}
.inputBox i{
    position: absolute;
    left: 0;
    bottom: 0;
    width: 100%;
    height: 2px;
    background: #45ff55;
    border-radius: 4px;
    transition: 0.5s;
    pointer-events: none;
    z-index: 0;
}
.inputBox input:valid ~ i,
.inputBox input:focus ~ i{
     height: 44px;
}

.links{
    display: flex;
    justify-content: space-between;

}

.links a {
    margin: 10px 0;
    font-size: 0.75em;
    color: #8f8f8f;
    text-decoration: none;
}
.links a:hover,
.links a:nth-child(2){
    color: #45ff55;
}

input[type="submit"]{
    border: none;
    outline: none;
    background: #45ff55;
    padding: 11px 25px;
    width: 100px;
    margin-top: 10px;
    border-radius: 4px;
    font-weight: 600;
    cursor: pointer;
}
input[type="submit"]:active{
    opacity: 0.8;
}
```

```
 .font{
   font-family: 'Poppins', sans-serif;
   top: 0;
   margin-left: 120%;
   text-align: left;
 }
 .fa{
   color:  aliceblue;



 }

 .navbar {
 overflow: hidden;
 background:linear-gradient(-13deg, transparent,#45ff55,#21e231);
 position: fixed;
 top: 0;
 width: 100%;
 font-size: 3vh;
}

.navbar a {
 float: right;
 display: block;
 color: #f2f2f2;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
 font-size: 17px;
}

.navbar a:hover {
 background: #ddd;
 color: black;
}
.title{
   padding: 12px 12px;
   color: #f2f2f2;
}

   </style>
</head>
<body>
   <div class="navbar">
       <a href="/login">SIGN IN </a>
       <a href="/register">SIGN UP</a>
       <a href="/adminlogin">ADMIN</a>
       <div class="title">
           <h3  style="right:0;font-family:'Lobster', cursive;">InfiniteArt</h3>
         </div>
     </div>



   <form method="post">
       <div class="box">
```

```
            <div class="form">
                <h2>SIGN IN</h2>

                <div class="inputBox"  >

                    <input type="text" name="email"   required>
                    <span  class="fa fa-envelope"> <span class="font">Email</span></span>
                    <i></i>
                </div>

                <div class="inputBox">
                    <input type="password" name="password" required>
                    <span  class="fa fa-lock"> <span class="font">Password</span></span>
                    <i></i>
                </div>
                <p>{{success}}</p>
                <p style="color: red">{{error}}</p>
                <div class="links">
                    <a href="#">Forget password</a>
                    <a href="/admin">SIGN UP</a>
                </div>
                <input type="submit" value="Login">
            </div>
        </div>
    </form>
</body>
</html>
```

adminregister.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Log in</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" href="{{url_for('static', filename='styles/style.css')}}">
</head>
<body>
    <div class="navbar">
        <a href="/login">SIGN IN </a>
        <a href="/register">SIGN UP</a>
        <a href="/adminlogin">ADMIN</a>
        <div class="title">
            <h3  style="right:0;font-family:'Lobster', cursive;">InfiniteArt</h3>
          </div>
      </div>
    <form method="post">
        <div class="box">
            <div class="form">
                <h2>SIGN UP</h2>

                <div class="inputBox"  >

                    <input type="text" name="username"   required>
                    <span  class="fa fa-user"> <span class="font">Username</span></span>
                    <i></i>
                </div>
                <div class="inputBox"  >

                    <input type="email" name="email"   required>
                    <span  class="fa fa-envelope"> <span class="font">Email</span></span>
                    <i></i>
                </div>
                <div class="inputBox"  >

                    <input type="text" name="phoneno"   required>
                    <span  class="fa fa-phone"> <span class="font">PhoneNumber</span></span>
                    <i></i>
                </div>
                <div class="inputBox">
                    <input type="password" name="password" required>
                    <span  class="fa fa-lock"> <span class="font">Password</span></span>
                    <i></i>
                </div>
                <p>{{success}}</p>
                <p style="color: red">{{error}}</p>
                <div class="links">
                    <a href="#">Forget password</a>
                    <a href="/adminlogin">SIGN IN</a>
                </div>
                <input type="submit" value="Login">
            </div>
        </div>
    </form>
</body>
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>View Product</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" href="{{url_for('static', filename='styles/style.css')}}">
<style>
    @import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,1
00;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');
                *{
                    font-family: 'Poppins', sans-serif;
                }

            body{
                    display: flex;
                    justify-content: center;
                    align-items: center;
                    min-height: 100vh;
                    background: #131313;
                }

            .container{
                    position: relative;
                }

            .container .card{
                    position: relative;
                    width: 320px;
                    height: 450px;
                    background: #232323;
                    border-radius: 20px;
                    overflow: hidden;
                 }

            .container .card:before{
                content: '';
                position: absolute;
                top: 0;
                left: 0;
                width: 100%;
                height: 100%;
                background: #45ff55;
                clip-path: circle(150px at 80% 20%);
                transition: 0.5s ease-in-out;
            }

            .container .card:hover:before{
                clip-path: circle(300px at 80% -20%);
            }

            .container .card:after{
                content: 'otto';
                position: absolute;
                top: 30%;
                left: -20%;
                font-size: 12em;
                font-weight: 800;
                font-style: italic;
                color: rgba(255,255,25,0.05)
            }

            .container .card .image{
                position: absolute;
                top: 50%;
                transform: translateY(-50%);
                z-index: 10000;
                width: 100%;
                height: 220px;
                transition: 0.5s;
            }

            .container .card:hover .image{
                top: 0%;
                transform: translateY(0%);
```

```
}

.container .card .image img{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    width: 150px;
    height: 170px;
}

.container .card .descrpton{
    position: absolute;
    bottom: 0;
    width: 100%;
    height: 100px;
    text-align: center;
    transition: 1s;
    z-index: 10;
}

.container .card:hover .descrpton{
     height: 210px;
}

.container .card .descrpton h2{
    position: relative;
    font-weight: 600;
    letter-spacing: 1px;
    color: #fff;
    margin: 0;
}

.container .card .descrpton .size, .container .card .descrpton .color {
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 8px 20px;
    transition: 0.5s;opacity: 0;
    visibility: hidden;
    padding-top: 0;
    padding-bottom: 0;
}

.container .card:hover .descrpton .size{
    opacity: 1;
    visibility: visible;
    transition-delay: 0.5s;
}

.container .card:hover .descrpton .color{
    opacity: 1;
    visibility: visible;
    transition-delay: 0.6s;
}
.container .card .descrpton .size h3, .container .card .descrpton .color h3{
    color: #fff;
    font-weight: 300;
    font-size: 14px;
    text-transform: uppercase;
    letter-spacing: 2px;
    margin-right: 10px;
}

.container .card .descrpton .size span{
    width: 26px;
    height: 26px;
    text-align: center;
    line-height: 26px;
    font-size: 14px;
    display: inline-block;
    color: #111;
    background: #fff;
    margin: 0 5px;
    transition: 0.5s;
    color: #111;
    border-radius: 4px;
    cursor: pointer;
}

.container .card .descrpton .size span:hover{
    background: #9bdc28;
}

.container .card .descrpton .color span{
    width: 20px;
    height: 20px;
    background: #ff0;
    border-radius: 50%;
    margin: 0 5px;
    cursor: pointer;
}

.container .card .descrpton .color span:nth-child(2){
    background: #9bdc28;
}

.container .card .descrpton .color span:nth-child(3){
    background: #03a9f4;
}

.container .card .descrpton .color span:nth-child(4){
    background: #e91e63;
}

.container .card .descrpton a{
    display: inline-block;
    padding: 10px 20px;
    background: #fff;
    border-radius: 4px;
    margin-top: 10px;
    text-decoration: none;
```

```
        font-weight: 600;
        color: #111;
        opacity: 0;
        transform: translateY(50px);
        transition: 0.5s;
        margin-top: 0;
      }

      .container .card:hover .descrpton a{
      opacity: 1;
      transform: translateY(0px);
      transition-delay: 0.75s;

      }




</style>

</head>
<body>
 <div class="navbar">
  <a href="/logout">LOG OUT</a>
  <div class="title">
    <h3  style="right:0;font-family:'Lobster', cursive;">InfiniteArt</h3>
   </div>
 </div>
  <div class="container">
    <div class="card">
      <div class="image">
        <img src="{{url_for('static',filename='shirt.jpeg')}}">
      </div>
      <div class="descrpton">
        <h2>SHIRT</h2>
        <div class="size">
          <span>7</span>
          <span>8</span>
          <span>9</span>
          <span>10</span>
        </div>
        <div class="color">
          <h3>Color: </h3>
          <span></span>
          <span></span>
          <span></span>
        </div>
        <a href="#">Buy now</a>
      </div>

      </div>
  </div>
</body>
</html>
```

index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <link href="//db.onlinewebfonts.com/c/d8a3c95906aec0c2483082a82e72cb40?family=WanderlustShine-Regular"
rel="stylesheet" type="text/css"/>


    <title>SMART FASHION RECOMMENDED APPLICATION</title>
    <style>
         @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');
         @import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
         @font-face{
            font-family: Wanderlust;
            src: url('/Wanderlust Letters-Font/OTF/WanderlustLetters-Regular.otf');
            font-weight: bold;}
*{
    margin: 0;
    padding: 0;

    font-family: 'Poppins', sans-serif;
}
body{
    text-align: center;

    background-color: #23242a;
    color: #f2f2f2;

}


          .navbar {
            overflow: hidden;


            position: fixed;
            top: 0;
            width: 100%;
            }
        .navbar a {
        float: right;
        display: block;
        color: #21e231;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
        font-size: 17px;
        }

        .navbar a:hover {
        background: #f2f2f2;
        color: black;
        }
```

```
.navbar h3{
    padding: 12px 12px;
    color: #21e231;
    font-size: 5vh;
    font-weight: 700;
    top:0;
    text-align: left;
}


.content {
width:100%;
height:100%;
display: table;
border:2px solid white;
}


.image{
    width:5%;
    height: 90%;
    float: left;
    padding: 2px;
}
.image img{
    height: 46px;
    width:80px;
}
.banner{
    width: 100%;
    height: 730px;
    background-image: url("{{url_for('static',filename='background.png')}}");
    background-attachment: fixed;
    background-size: cover;
}
.con{
    text-align: center;
    padding: 2%;
}
.con a{
    color: #f2f2f2;
    font-family: 'Poppins', sans-serif;
    font-weight: bolder;
    text-decoration: none;
    background: #21e231;
    padding: 5px;
    border-radius: 10px;

}
.con a:hover{
    background:#f2f2f2;
    color: #21e231;
    cursor: pointer;
    font-family: 'Poppins', sans-serif;
    font-weight: bolder;
}
```

```
        .get{
            margin: 20% 20%;
        }
        .about,.cart,.chatbot{
            margin: 2%;
            width: 90%;
            height: 300px;
            box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
            background-color: #111111;
            transition: 0.3s;
            padding: 1%;
        }
        .about:hover,.cart:hover,.chatbot:hover {
            box-shadow: 0 8px 16px 0 rgb(52, 230, 28);
            }

        .left{

            width: 10%;
            height: 100%;
            float: left;
            margin: 1%;



        }
        .about img,.chatbot img{
            width: 20%;
            height: 85%;
            float: left;
            top: 0;
            margin: 0% 6%;
        }
        .cart img{
            width: 20%;
            height: 85%;
            float: right;
            top: 0;
            margin: 0% 6%;
        }
        .txt{

            width: 55%;
            float: right;
            text-align: justify;
            margin: 4% 5%;


        }

    </style>

</head>
<body>
        <div class="navbar">
            <a href="/login">SIGN IN </a>
```

```
          <div class="title">
         <!-- <div class="image"> <img src="{{url_for('static', filename='logo.png')}}"></div> -->
          <div><h3 style="right:0;font-family:'Lobster', cursive;">InfiniteArt </h3></div>
          </div>

      </div>
    <div class="banner">
      <div class="con">

      <div class="get"><h1><span style="color:#21e231; ">SMART FASHION RECOMMENDER APPLICATION,</span>
unlimitted access to all fashion items</h1><br>
          <a href="/register"  name="get started">GET STARTED &#8594;</a></div>
     </div>
    </div>
    <br>
    <div class="about">
      <center><h2>ABOUT</h2></center>
           <img src="{{url_for('static',filename='about.png')}}">

              <div class="txt">
                  <p><h3 style="color:#21e231;">A warm welcome to your online fashion lane.</h3>

                  We are aware of how much you value fashion and how much you enjoy online shopping's convenience. When
you're on the go, the InfiniteArt online shopping app makes sure you don't miss out on a chic shopping experience. If you
want to shop for clothing for men and women, shoes, accessories, or even the newest electronics and tech gadgets,
InfiniteArt is always there to meet all of your style needs
              </p></div>

    </div>
    <div class="cart">
      <center><h2>RECOMMENDATION&numsp;SYSTEM</h2></center>
           <img src="{{url_for('static',filename='cart.png')}}">

              <div class="txt">
                  <p><h3 style="color:#21e231;">Your Favourite Items are here.</h3>

                      Here, we used a recommendation system that is to categorise the user's clothing and suggest the best outfit
for a particular occasion based on a recommendation algorithm. The suggested system demonstrates that it can analyse the
user's attire from the images, determine the type and colour of the outfit, and then suggest the most appropriate outfit for the
situation based on the user's current attire.
              </p></div>

    </div>
    <div class="chatbot">
      <center><h2>CHATBOT</h2></center>
           <img src="{{url_for('static',filename='chatbot.png')}}">

              <div class="txt">
                  <p><h3 style="color:#21e231;">Your Heleper Is Here.</h3>

                      Chatbots can also be used to gather visitor data, which can then be used to improve product
recommendations and suggestions. You can personalise product pages and increase customer loyalty and affinity by having
a thorough understanding of customer inquiries, needs, and preferences. Chatbots can also inform customers when an item
is out of stock, suggest suitable alternatives based on their preferences, and let them know when their order is expected to
arrive.
              </p></div>
      </div>
      <div class="devlopers"></div>
</body>
</html><!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <link href="//db.onlinewebfonts.com/c/d8a3c95906aec0c2483082a82e72cb40?family=WanderlustShine-Regular"
rel="stylesheet" type="text/css"/>


    <title>SMART FASHION RECOMMENDED APPLICATION</title>
    <style>
         @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');
         @import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
         @font-face{
            font-family: Wanderlust;
            src: url('/Wanderlust Letters-Font/OTF/WanderlustLetters-Regular.otf');
            font-weight: bold;}
*{
    margin: 0;
    padding: 0;

    font-family: 'Poppins', sans-serif;
}
body{
    text-align: center;

    background-color: #23242a;
    color: #f2f2f2;

}



          .navbar {
            overflow: hidden;


            position: fixed;
            top: 0;
            width: 100%;
            }
        .navbar a {
        float: right;
        display: block;
        color: #21e231;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
        font-size: 17px;
        }
```

```
.navbar a:hover {
background: #f2f2f2;
color: black;
}
.navbar h3{
    padding: 12px 12px;
    color: #21e231;
    font-size: 5vh;
    font-weight: 700;
    top:0;
    text-align: left;
}


.content {
width:100%;
height:100%;
display: table;
border:2px solid white;
}


.image{
    width:5%;
    height: 90%;
    float: left;
    padding: 2px;
}
.image img{
    height: 46px;
    width:80px;
}
.banner{
    width: 100%;
    height: 730px;
    background-image: url("{{url_for('static',filename='background.png')}}");
    background-attachment: fixed;
    background-size: cover;
}
.con{
    text-align: center;
    padding: 2%;
}
.con a{
    color: #f2f2f2;
    font-family: 'Poppins', sans-serif;
    font-weight: bolder;
    text-decoration: none;
    background: #21e231;
    padding: 5px;
    border-radius: 10px;

}
.con a:hover{
    background:#f2f2f2;
```

```
        color: #21e231;
        cursor: pointer;
        font-family: 'Poppins', sans-serif;
        font-weight: bolder;
    }
    .get{
        margin: 20% 20%;
    }
    .about,.cart,.chatbot{
        margin: 2%;
        width: 90%;
        height: 300px;
        box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
        background-color: #111111;
        transition: 0.3s;
        padding: 1%;
    }
    .about:hover,.cart:hover,.chatbot:hover {
        box-shadow: 0 8px 16px 0 rgb(52, 230, 28);
        }

    .left{

        width: 10%;
        height: 100%;
        float: left;
        margin: 1%;



    }
    .about img,.chatbot img{
        width: 20%;
        height: 85%;
        float: left;
        top: 0;
        margin: 0% 6%;
    }
    .cart img{
        width: 20%;
        height: 85%;
        float: right;
        top: 0;
        margin: 0% 6%;
    }
    .txt{

        width: 55%;
        float: right;
        text-align: justify;
        margin: 4% 5%;


    }

</style>
```

```
</head>
<body>
      <div class="navbar">
          <a href="/login">SIGN IN </a>
          <div class="title">
         <!-- <div class="image"> <img src="{{url_for('static', filename='logo.png')}}"></div> -->
          <div><h3 style="right:0;font-family:'Lobster', cursive;">InfiniteArt </h3></div>
          </div>

      </div>
    <div class="banner">
      <div class="con">

      <div class="get"><h1><span style="color:#21e231; ">SMART FASHION RECOMMENDER APPLICATION,</span>
unlimitted access to all fashion items</h1><br>
          <a href="/register"  name="get started">GET STARTED &#8594;</a></div>
     </div>
    </div>
    <br>
    <div class="about">
      <center><h2>ABOUT</h2></center>
           <img src="{{url_for('static',filename='about.png')}}">

              <div class="txt">
                  <p><h3 style="color:#21e231;">A warm welcome to your online fashion lane.</h3>

                  We are aware of how much you value fashion and how much you enjoy online shopping's convenience. When
you're on the go, the InfiniteArt online shopping app makes sure you don't miss out on a chic shopping experience. If you
want to shop for clothing for men and women, shoes, accessories, or even the newest electronics and tech gadgets,
InfiniteArt is always there to meet all of your style needs
              </p></div>

    </div>
    <div class="cart">
      <center><h2>RECOMMENDATION&numsp;SYSTEM</h2></center>
           <img src="{{url_for('static',filename='cart.png')}}">

              <div class="txt">
                  <p><h3 style="color:#21e231;">Your Favourite Items are here.</h3>

                      Here, we used a recommendation system that is to categorise the user's clothing and suggest the best outfit
for a particular occasion based on a recommendation algorithm. The suggested system demonstrates that it can analyse the
user's attire from the images, determine the type and colour of the outfit, and then suggest the most appropriate outfit for the
situation based on the user's current attire.
              </p></div>

    </div>
    <div class="chatbot">
      <center><h2>CHATBOT</h2></center>
           <img src="{{url_for('static',filename='chatbot.png')}}">

              <div class="txt">
                  <p><h3 style="color:#21e231;">Your Heleper Is Here.</h3>

                      Chatbots can also be used to gather visitor data, which can then be used to improve product
```

recommendations and suggestions. You can personalise product pages and increase customer loyalty and affinity by having a thorough understanding of customer inquiries, needs, and preferences. Chatbots can also inform customers when an item is out of stock, suggest suitable alternatives based on their preferences, and let them know when their order is expected to arrive.

```
                </p></div>

      </div>
      <div class="devlopers"></div>
</body>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Log in</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <style>
         @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');
         @import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
body{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-color: #23242a;

}
.box{
    position: relative;
    width: 380px;
    height: 400px;
    background-color: #1c1c1c;
    border-radius: 8px;
    overflow: hidden;
}
.box::before{
    content: '';
    position: absolute;
    top: -50%;
    left: -50%;
    width: 380px;
    height: 420px;
    background: linear-gradient(0deg, transparent,#45ff55,#45ff55);
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
```

```
}
.box::after{
    content: '';
    position: absolute;
    top: -50%;
    left: -50%;
    width: 380px;
    height: 420px;
    background: linear-gradient(0deg, transparent,#45ff55,#45ff55);
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
    animation-delay: -3s;
}
@keyframes animate {
    0%{
        transform: rotate(0deg);
    }
    100%{
        transform: rotate(350deg);
    }
}

/* Form */
.form{
    position: absolute;
    inset: 2px;
    border-radius: 8px;
    background: #28292d;
    z-index: 10;
    padding: 40px 40px;
    display: flex;
    flex-direction: column;
}
.form h2{
    color: #45ff55;
    font-weight: 500;
    text-align: center;
    letter-spacing: 0.1em;
}
.inputBox{
    position: relative;
    width: 100%;
    margin-top: 35px;
}

.inputBox input{
    position: relative;
    width: 100%;
    padding: 10px 10px 10px;
    background: transparent;
    border: none;
    outline: none;
    color: #23242a;
    font-size: 1em;
    letter-spacing: 0.05em;
    z-index: 10;
```

```css
}
.inputBox span{
    position: absolute;
    left: 0;
    padding: 20px 0px 10px;
    font-size: 1em;
    color: white;
    pointer-events: none;
    letter-spacing: 0.05em;
    transition: 0.5s;
}
.inputBox input:valid ~ span,
.inputBox input:focus ~ span{
    color: #45ff55;
    transform: translateX(0px) translateY(-34px);
    font-size: 0.75em;
}
.inputBox i{
    position: absolute;
    left: 0;
    bottom: 0;
    width: 100%;
    height: 2px;
    background: #45ff55;
    border-radius: 4px;
    transition: 0.5s;
    pointer-events: none;
    z-index: 0;
}
.inputBox input:valid ~ i,
.inputBox input:focus ~ i{
     height: 44px;
}

.links{
    display: flex;
    justify-content: space-between;

}

.links a {
    margin: 10px 0;
    font-size: 0.75em;
    color: #8f8f8f;
    text-decoration: none;
}
.links a:hover,
.links a:nth-child(2){
    color: #45ff55;
}

input[type="submit"]{
    border: none;
    outline: none;
    background: #45ff55;
    padding: 11px 25px;
    width: 100px;
    margin-top: 10px;
    border-radius: 4px;
    font-weight: 600;
    cursor: pointer;
}
input[type="submit"]:active{
    opacity: 0.8;
}



  .font{
    font-family: 'Poppins', sans-serif;
    top: 0;
    margin-left: 120%;
    text-align: left;
  }
  .fa{
    color:  aliceblue;



  }

  .navbar {
  overflow: hidden;
  background:linear-gradient(-13deg, transparent,#45ff55,#21e231);
  position: fixed;
  top: 0;
  width: 100%;
  font-size: 3vh;
}

.navbar a {
  float: right;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}

.navbar a:hover {
  background: #ddd;
  color: black;
}
.title{
    padding: 12px 12px;
    color: #f2f2f2;
}

    </style>
</head>
<body>
    <div class="navbar">
        <a href="/login">SIGN IN </a>
        <a href="/register">SIGN UP</a>
        <a href="/adminlogin">ADMIN</a>
        <div class="title">
            <h3  style="right:0;font-family:'Lobster', cursive;">InfiniteArt</h3>
          </div>
      </div>


    <form method="post">
        <div class="box">
            <div class="form">
                <h2>SIGN IN</h2>

                <div class="inputBox"  >

                    <input type="text" name="email"   required>
                    <span  class="fa fa-envelope"> <span class="font">Email</span></span>
                    <i></i>
                </div>

                <div class="inputBox">
                    <input type="password" name="password" required>
                    <span  class="fa fa-lock"> <span class="font">Password</span></span>
                    <i></i>
                </div>
                <p>{{success}}</p>
                <p style="color: red">{{error}}</p>
                <div class="links">
                    <a href="#">Forget password</a>
                    <a href="/register">SIGN UP</a>
                </div>
                <input type="submit" value="Login">
            </div>
        </div>
    </form>
</body>
</html>
```

register html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Log in</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" href="{{url_for('static', filename='styles/style.css')}}">
</head>
<body>
    <div class="navbar">
        <a href="/login">SIGN IN </a>
        <a href="/register">SIGN UP</a>
        <a href="/adminlogin">ADMIN</a>
        <div class="title">
            <h3  style="right:0;font-family:'Lobster', cursive;">InfiniteArt</h3>
```

```html
     </div>
   </div>
  <form method="post">
    <div class="box">
      <div class="form">
        <h2>SIGN UP</h2>

        <div class="inputBox"  >

          <input type="text" name="username"   required>
          <span  class="fa fa-user"> <span class="font">Username</span></span>
          <i></i>
        </div>
        <div class="inputBox"  >

          <input type="email" name="email"   required>
          <span  class="fa fa-envelope"> <span class="font">Email</span></span>
          <i></i>
        </div>
        <div class="inputBox"  >

          <input type="text" name="phoneno"   required>
          <span  class="fa fa-phone"> <span class="font">PhoneNumber</span></span>
          <i></i>
        </div>
        <div class="inputBox">
          <input type="password" name="password" required>
          <span  class="fa fa-lock"> <span class="font">Password</span></span>
          <i></i>
        </div>
        <p>{{success}}</p>
        <p style="color: red">{{error}}</p>
        <div class="links">
          <a href="#">Forget password</a>
          <a href="/login">SIGN IN</a>
        </div>
        <input type="submit" value="Login">
      </div>
    </div>
  </form>
</body>
</html>
```

style.css

```css
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');
 @import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
```

```css
    background-color: #23242a;

}
.box{
    position: relative;
    width: 380px;
    height: 530px;
    background-color: #1c1c1c;
    border-radius: 8px;
    overflow: hidden;
}
.box::before{
    content: '';
    position: absolute;
    top: -50%;
    left: -50%;
    width: 380px;
    height: 420px;
    background: linear-gradient(0deg, transparent,#45ff55,#45ff55);
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
}
.box::after{
    content: '';
    position: absolute;
    top: -50%;
    left: -50%;
    width: 380px;
    height: 420px;
    background: linear-gradient(0deg, transparent,#45ff55,#45ff55);
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
    animation-delay: -3s;
}
@keyframes animate {
    0%{
        transform: rotate(0deg);
    }
    100%{
        transform: rotate(350deg);
    }
}

/* Form */
.form{
    position: absolute;
    inset: 2px;
    border-radius: 8px;
    background: #28292d;
    z-index: 10;
    padding: 40px 40px;
    display: flex;
    flex-direction: column;
}
.form h2{
    color: #45ff55;
    font-weight: 500;
    text-align: center;
    letter-spacing: 0.1em;
}
.inputBox{
    position: relative;
    width: 100%;
    margin-top: 35px;
}

.inputBox input{
    position: relative;
    width: 100%;
    padding: 10px 10px 10px;
    background: transparent;
    border: none;
    outline: none;
    color: #23242a;
    font-size: 1em;
    letter-spacing: 0.05em;
    z-index: 10;
}
.inputBox span{
    position: absolute;
    left: 0;
    padding: 20px 0px 10px;
    font-size: 1em;
    color: white;
    pointer-events: none;
    letter-spacing: 0.05em;
    transition: 0.5s;
}
.inputBox input:valid ~ span,
.inputBox input:focus ~ span{
    color: #45ff55;
    transform: translateX(0px) translateY(-34px);
    font-size: 0.75em;
}
.inputBox i{
    position: absolute;
    left: 0;
    bottom: 0;
    width: 100%;
    height: 2px;
    background: #45ff55;
    border-radius: 4px;
    transition: 0.5s;
    pointer-events: none;
    z-index: 0;
}
.inputBox input:valid ~ i,
.inputBox input:focus ~ i{
     height: 44px;
}

.links{
    display: flex;
    justify-content: space-between;

}

.links a {
    margin: 10px 0;
    font-size: 0.75em;
    color: #8f8f8f;
    text-decoration: none;
}
.links a:hover,
.links a:nth-child(2){
    color: #45ff55;
}

input[type="submit"]{
    border: none;
    outline: none;
    background: #45ff55;
    padding: 11px 25px;
    width: 100px;
    margin-top: 10px;
    border-radius: 4px;
    font-weight: 600;
    cursor: pointer;
}
input[type="submit"]:active{
    opacity: 0.8;
}


  .font{
    font-family: 'Poppins', sans-serif;
    top: 0;
    margin-left: 120%;
    text-align: left;
  }
  .fa{
    color:  aliceblue;



  }



  .navbar {
    overflow: hidden;
    background:linear-gradient(-13deg, transparent,#45ff55,#21e231);
    position: fixed;
    top: 0;
    width: 100%;
    font-size: 3vh;
  }

  .navbar a {
    float: right;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    font-size: 17px;
  }

  .navbar a:hover {
    background: #ddd;
    color: black;
  }
  .title{
      padding: 12px 12px;
      color: #f2f2f2;
  }
```

Flask.py

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import bcrypt
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURITY=SSL;
SSLServerCertificateDigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=nhl80748;PWD=3yD0G9e6VuQHsOBX;", "", "")
#url_for('static', filename='style.css')


app = Flask(__name__)
app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'

@app.route("/",methods=['GET'])
def home():
    if 'email' not in session:
      return redirect(url_for('index'))
    return render_template('home.html',name='Home')
@app.route("/index")
def index():
  return render_template('index.html')

@app.route("/register",methods=['GET','POST'])
def register():
  if request.method == 'POST':
    username = request.form['username']
    email = request.form['email']
    phoneno = request.form['phoneno']
    password = request.form['password']

    if not username or not email or not phoneno or not password:
      return render_template('register.html',error='Please fill all fields')
    hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())
    query = "SELECT * FROM user_detail WHERE email=? OR phoneno=?"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.bind_param(stmt,2,phoneno)
    ibm_db.execute(stmt)
    isUser = ibm_db.fetch_assoc(stmt)
    if not isUser:
      insert_sql = "INSERT INTO user_detail(username, email, phoneno, password) VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, phoneno)
      ibm_db.bind_param(prep_stmt, 4, hash)
      ibm_db.execute(prep_stmt)
      return render_template('register.html',success="You can login")
    else:
      return render_template('register.html',error='Invalid Credentials')

  return render_template('register.html',name='Home')

@app.route("/login",methods=['GET','POST'])
def login():
    if request.method == 'POST':
      email = request.form['email']
      password = request.form['password']

      if not email or not password:
        return render_template('login.html',error='Please fill all fields')
      query = "SELECT * FROM user_detail WHERE email=?"
      stmt = ibm_db.prepare(conn, query)
      ibm_db.bind_param(stmt,1,email)
      ibm_db.execute(stmt)
      isUser = ibm_db.fetch_assoc(stmt)
      print(isUser,password)

      if not isUser:
        return render_template('login.html',error='Invalid Credentials')

      isPasswordMatch = bcrypt.checkpw(password.encode('utf-8'),isUser['PASSWORD'].encode('utf-8'))

      if not isPasswordMatch:
        return render_template('login.html',error='Invalid Credentials')

      session['email'] = isUser['EMAIL']
      return redirect(url_for('home'))

    return render_template('login.html',name='Home')

@app.route("/admin",methods=['GET','POST'])
def adregister():
  if request.method == 'POST':
    username = request.form['username']
    email = request.form['email']
    phoneno = request.form['phoneno']
    password = request.form['password']

    if not username or not email or not phoneno or not password:
      return render_template('adminregister.html',error='Please fill all fields')
    hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())
    query = "SELECT * FROM admin_detail WHERE email=? OR phoneno=?"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.bind_param(stmt,2,phoneno)
    ibm_db.execute(stmt)
    isUser = ibm_db.fetch_assoc(stmt)
    if not isUser:
      insert_sql = "INSERT INTO admin_detail(username, email, phoneno, password) VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, phoneno)
      ibm_db.bind_param(prep_stmt, 4, hash)
      ibm_db.execute(prep_stmt)
      return render_template('adminregister.html',success="You can login")
    else:
      return render_template('adminregister.html',error='Invalid Credentials')

  return render_template('adminregister.html',name='Home')

@app.route("/adminlogin",methods=['GET','POST'])
def adlogin():
    if request.method == 'POST':
      email = request.form['email']
      password = request.form['password']

      if not email or not password:
        return render_template('adminlogin.html',error='Please fill all fields')
      query = "SELECT * FROM admin_detail WHERE email=?"
      stmt = ibm_db.prepare(conn, query)
      ibm_db.bind_param(stmt,1,email)
      ibm_db.execute(stmt)
      isUser = ibm_db.fetch_assoc(stmt)
      print(isUser,password)

      if not isUser:
        return render_template('adminlogin.html',error='Invalid Credentials')

      isPasswordMatch = bcrypt.checkpw(password.encode('utf-8'),isUser['PASSWORD'].encode('utf-8'))

      if not isPasswordMatch:
        return render_template('adminlogin.html',error='Invalid Credentials')

      session['email'] = isUser['EMAIL']
      return redirect(url_for('home'))

    return render_template('adminlogin.html',name='Home')

@app.route('/logout')
def logout():
    session.pop('email', None)
    return redirect(url_for('login'))
if __name__ == "__main__":
    app.run(debug=True)
```

Github link:

https://github.com/IBM-EPBL/IBM-Project-47898-1660803141.git

Demo link:

https://drive.google.com/file/d/16R5w2g2dtdQg2a-QyMPnXEVTxc0w11Rk/view?usp=drivesdk

DISCUSSION

This scholarly article has provided a comprehensive review of the methods, algorithmic models and filtering techniques used in the recent fashion recommendation-based research papers. However, this review paper has some limitations too. Primarily, the focus of this comprehensive review paper was to explore fashion recommendation-based articles published in last decade the explicitly described their frameworks, algorithms, and filtering techniques. To achieve this goal, the articles were searched using keywords relevant to the topic title instead of using the PRISMA technique. However, it did not affect the article extraction methodology, because the authors included and studied all the research papers relevant to the research focus. However, future researchers could conduct a systematic literature review on the same topic. The initial keyword searching did not include “garment” and “outfit”; however, this did not influence the search results because we also studied the fashion recommendation articles that contained these keywords. The future research can also conduct a review of the datasets that have been used in fashion recommendation-based research articles. Additionally, further reviews of fashion recommendation systems can apply our proposed potential algorithms to any of the available fashion image datasets to evaluate the performance of the recommender systems.

CONCLUSION

Recommendaton systems have the potental to explore new opportunites for retailers by enabling them to provide customized recommendatons to consumers based on informaton retrieved from the Internet. They help consumers to instantly fnd the products and services that closely match with their choices. Moreover, diferent stat-of-the-art algorithms have been developed to recommend products based on users' interactons with their social groups. Therefore, research on embedding social media images within fashion recommendaton systems has gained huge popularity in recent tmes. This paper presented a review of the fashion recommendaton systems, algorithmic models and fltering techniques based on the academic artcles related to this topic. The technical aspects, strengths and weaknesses of the fltering techniques have been discussed elaborately, which will help future researchers gain an in- depth understanding of fashion recommender systems. However, the proposed prototypes should be tested in commercial applicatons to understand their feasibility and accuracy in the retail market, because inaccurate recommendatons can produce a negatve impact on a customer. Moreover, future research should concentrate on including tme series analysis and accurate categorizaton of product images based on the variaton in color, trend and clothing style in order to develop an effective recommendation system

.

FUTURE SCOPE

Online selling and purchasing ofer innumerable benefts to both sellers and buyers, and these advantages are also the reasons for the rising scope of eCommerceWell, to put it bluntly, the scope of e-business in the near future looks to be ever-increasing and growing, because the trend has really caught on here. E-commerce giant Amazon is keen to conquer the Indian market and has already invested a great deal, especially with its 49% stake in the Future Group.

Indian online retail giant Flipkart has already opened a few ofine stores and plans more stores in smaller cites. They plan to combine online and ofine stores to maximize their selling potental.Google and Tata Trust have launched a joint program 'Saathi' to increase internet and mobile penetraton among rural women.The Government of India is also making a huge push for Ecommerce by providing numerous sops to startups, cyberparks, and so on through its Digital India program.As of now, there are close to 20,000 E-commerce companies in India, with many more expected to join the bandwagon every month.

REFERENCE

1. Guan, C.; Qin, S.; Ling, W.; Ding, G. Apparel recommendaton system evoluton: An empirical review. Int. J. Cloth. Sci. Technol. 2016, 28, 854–879, doi:10.1108/ijcst-09-20150100.
2. Hu, Y.; Manikonda, L.; Kambhampat, S. What we Instagram: A frst analysis of Instagram photo content and user types. Available online: htp://www.aaai.org (accessed on 1 May 2014).
3. Gao, G.; Liu, L.; Wang, L.; Zhang, Y. Fashion clothes matching scheme based on Siamese Network and AutoEncoder. Multmed. Syst. 2019, 25, 593–602, doi:10.1007/s00530-01900617-9.
4. Liu, Y.; Gao, Y.; Feng, S.; Li, Z. Weather-to-garment: Weather-oriented clothing recommendaton. In Proceedings of the 2017 IEEE Internatonal Conference on

Multmedia and Expo. (ICME), Hong Kong, China, 31 August 2017; pp. 181–186, doi:10.1109/ICME.2017.8019476.

5. Chakraborty, S.; Hoque, M.S.; Surid, S.M. A comprehensive review on imagebased style predicton and online fashion recommendaton. J. Mod. Tech. Eng. 2020, 5, 212–233.