# AN INTELLIGENT FASHION RECOMMENDATION SYSTEM

Erandi Praboda Hettiarachchi

(199328R)

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

April 2021

# AN INTELLIGENT FASHION RECOMMENDATION SYSTEM

Erandi Praboda Hettiarachchi

(199328R)

This dissertation submitted in partial fulfillment of the requirements for the degree

Master of Science in Computer Science specializing in Software Architecture

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

April 2021

# DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters thesis under my supervision.

Name of the supervisor: Dr. Indika Perera

Signature of the supervisor: Date

# ABSTRACT

In the present as a result of the rapid rising in living standards, most people have been tempted to develop their interest in shopping. As a result of that, nowadays there is a huge demand for garments and the number of people who pursue fashion has increased. Since there are different types of garments available in the market people have to figure out what needs to buy and this will lead to trying garments repeatedly, which consumes more time for the selection. On the other hand, even though most of the sellers have online stores the real benefits of online stores cannot be obtained because the consumers always have a doubt whether the purchase will be matching with him or her. Besides all of these, it is somewhat impossible for the merchant to identify the real customer demand and create an outfit based on each person's satisfaction. In the present most of the available recommendation systems will recommend the clothes for a user based on the activities or the behavior of the other users who used the system previously by considering that all users behave simultaneously. The current user's personal preferences will not take into account. But when it comes to the cloth recommendation, it is very crucial to consider the users' personal preference. This paper presented an automated way of recommending outfits based on the user image by directly incorporating the visual signals into the recommendation objective. This research provides more insights on how convolutional neural networks can be used for the feature extraction phase from fashion images and evaluate the output from different CNNs. To achieve better results than the available neural network this research is proposing a hybrid approach of using both ResNet and VGG. The final evaluation of the system proves that the hybrid approach is having a positive impact on achieving more accurate results than the existing systems.

# ACKNOWLEDGEMENTS

My profound gratitude goes to Dr. Indika Perera, my supervisor for the knowledge, supervision, advice, and guidance provided with his expertise, throughout making the thesis a success.

My appreciation goes to my family for the motivation and support provided throughout my life.

I also would like to thank my colleagues in the MSc batch and at my workplace for the help and support provided in managing my research work.

# TABLE OF CONTENT

## LIST OF FIGURES

## LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

Recommending a complete outfit based on user preferences is a problem and it can be challenging even for the best experts. Along with the development of the field of Information Technology and increasing computing power, the significance of E-commerce systems has been increased and various kinds of E-commerce systems play an important role in our day-to-day life. In the present people are tempted to purchase a lot of things online since it has been the easiest way to save time. When purchasing items from an online store it is very normal to see that the Systems will always recommend more compatible items that will be similar to the item that is purchasing or the items that are bought together by other customers. This way of recommending products will definitely enhance the consumer's purchase experience and increase the number of sales.

This traditional way of recommendation systems lies in the concept of collaborative filtering and content-based filtering where collaborative filtering, which filters out the items according to the user's buying behavior, and content-based filtering is built around the existing attributes available in products. Most of these traditional recommendation methods are required to use frequent item mining as the main part of the implementation[1]. When the recommendation systems are using collaborative filtering both of these methods require a large number of historical purchasing records to obtain more accurate results. But it will be really difficult to provide accurate results for new items since at the beginning there is no historical data available, this is called the "cold problem of start"[2][3]. Most of the related research work has been done with a considerable amount of investigations with the purpose of overcoming this problem by using content-based filtering. Other than content-based filtering the other possible way to solve this problem is the fine-grained analysis of attributes and subcategories by constructing the relationships that statistically prove which subcategories or which sets of attributes match each other. Even though it seems that this approach works well theoretically it is necessary to collect large-scale

datasets along with fine-grained categorical labels and domain knowledge for user experience. This method is not well generalized to work when adding new subcategories. With relation to the clothing, collecting a large data set will become difficult as the fashion collections change every season.
In order to overcome the dynamic behavior of the fashion industry and the difficulty of achieving higher customer satisfaction, this paper presents an intelligent fashion recommendation system that is capable of doing large-scale visual recommendations based on the given input image.

The term visual recommendation model is used to introduce a model which has the capability of incorporating the visual signals into the recommendation objective. Here the visual signals refer to the visual features extracted from the image and these features will be used by the recommendation system to guide the users' decisions. Most of the time users who are interested in buying particular items from online stores would be required to explore more visually similar items such as items with similar patterns, materials, colors, and shapes. The proposed system will provide the platform for that.

## 1.1 Problem

As mentioned earlier, with the rapid growth of information technology, online shopping has been more popular among the young generation. People would like to purchase the items virtually rather than stepping into the stores since nowadays the fashion industry has a huge collection of garments. It may be difficult to select the best one among the others.  Facing too many types of garments brings the necessity of trying them repeatedly; this will involve wasting considerable amounts of time and energy.  Besides, it would be not easy for merchants to achieve the actual demand of consumers with high customer satisfaction. Traditional garment recommendation would be based on manual operations, Therefore, it is required and meaningful to develop a fashion recommendation system that finds a set of visual objective indicators to indicate consumer's fashion interest, instead of subjective opinions.

Most of the time people would have images of fashion items that they are interested in and it would be easy for them if there is a way to search on the online store by just uploading these images rather than going through the whole catalog one by one. Plugging image-based searching on traditional online stores will require modification to their website and implementing a model which is capable of doing visual recommendation more accurately. In-house implementation of that kind of recommendation system will be expensive.

Therefore, the main research problem that is addressed through this research is designing and developing intelligent software that recommends fashion items with similar colors, patterns, and shapes to the given input fashion image.

## 1.2 Objectives

Gather knowledge from the existing researches, related to fashion recommendations and identify the problem areas that are not covered by the existing researches, and contribute to overcome these problems.

Apply the concepts in computer science and software architecture to provide a better user experience and achieve more accurate results.

The main objective of this research work would be designing and developing an intelligent fashion recommendation system that identifies each user's fashion interest more accurately and made recommendations to achieve higher user satisfaction.

## 1.3 Scope

At the completion of the research, an intelligent fashion recommendation system would be proposed and implemented which achieves better accuracy and performance. The proposed system provides the capabilities to the user to upload a fashion image to the system and obtain recommendations based on the uploaded image. The system will extract the most important features from the given images like the patterns, category, type of materials, and colors and provide recommendations based on the extracted basic and advanced features of the fashion image.

The scope of this research includes designing and implementing an intelligent fashion recommendation system by addressing the problems of the existing fashion recommendation systems. Other than that it also includes the creation of the most suitable data set, suggesting the techniques to preprocess the data set, selecting the most suitable conventional neural networks to build a feature extractor for the fashion image, training the final mode, and finally evaluating the model's results with proper matrices.

### 1.4 Outline

The first chapter discusses the research problem that would be addressed through this research. It also provided the motivation and the objectives that needed to be achieved during the completion of the research. The second chapter presents the other researches that have been done under the fashion recommendation and image processing domain. These two areas were selected to gather the current knowledge regarding the existing tools in the field of fashion recommendation systems. The methodology chapter presents the approaches that were followed during the design of the system. Several steps have been defined when designing and implementing the final proposed system. The next chapter discusses how the architecture of the proposed system has been designed and organized. The evaluation chapter presents the evaluation criteria used to evaluate the prototype system that was implemented under this research. And finally, the conclusion chapter discusses research contribution, limitations of the research, and the future work that can be done under this research

# CHAPTER 2

# LITERATURE REVIEW

Available researches on intelligent fashion designing were analyzed to understand the research background and their proposed solution under this domain. The approaches followed in those researches have given the direction to the methodology proposed through this research.

## 2. 1 Research carried under the fashion recommendation.

Even in the present, the clothing recommendation can be considered as a new research topic, which has only a few related kinds of literature. In 2011, Iwata et al. [4] proposed a personalized cloth recommendation system that overcomes the difficulty of gaining domain knowledge in the clothing domain as the fashion collections change very frequently over time. This proposed system uses the fully-body photographs available in magazines to do the coordination among available clothes. Researchers have used a topic model along with the regional features images to identify the intrinsic relationships between various fashion items. This proposed recommendation system mainly focuses on how it is composed of. To get the best results in this method, they have suggested using only the images which contain the entire human body, in other words, a complete dress with all clothing pieces. At the learning stage, they have used a selected set of images as the training data based on their visual characteristics. Other than that more visual details like textures and colors have been used to enhance the accuracy of the learning process. In this model, it is necessary to identify the lower and upper parts of a given outfit accurately. As the input of this system user has to provide an image, the matching clothing items for this given input will be recommended by the system. Individual clothing pieces are recognized by the system itself by their categories and attributes.

In a different study, Veit et al. [5] presented another way that is capable of transforming features from the cloth item images to the style space which expresses the compatibility. In their system, they used a Siamese Convolutional Neural

Network (CNN) [5] architecture for the image transformation where the training data set of the neural network were organized as pairs that are either compatible or incompatible. Recently McAuley et al. proposed a new methodology to represent the human sense of making connections among physical objects mainly based on their physical shape and appearance [6]. They presented it as a network interference problem defined with graphs of correlated images. This proposed system has proven its capability of recommending clothes and accessories which match each other.

An intelligent system called Magic Mirror was cast by Lui Si et al as a new approach for occasion cloth recommendation [7]. This system focused on modeling the different kinds of human body features such as low-level body features, middle-level facial, and body features. Based on the events and the activities that you wish to participate in, Magic Mirror will automatically recommend the most suitable dressing style, hairstyle, and makeup. Therefore the user will be able to present themselves with an elegant appearance within a small period of time. The researchers have adapted a latent support vector machine-based model for the deciding matching rules for the recommendation system.

In 2017, Y. Li et al. presented a method to Mining Fashion Outfit Composition Using an End-to-End Deep Learning Approach on Set Data. The proposed system was a machine learning approach to compose an outfit automatically. The core of this proposed system was to score the fashion outfit candidates based on their appearance and metadata [8].

### 2.2 Research carried under Image processing.

Detecting faces from a color image can be categorized as a critical face in an identification application because detecting skin color from the complex background adds more complexities to this. Most of the available related literature has overcome this problem by assuming that the location of the face is already known in advance. Most of these approaches utilized techniques such as Neural Network, Machine Learning, Principal Component Analysis, Information theory, Geometrical Modeling, Template Matching, Color analysis, etc.

A. S. Tolba et al have presented a literature review of the most recent face recognition algorithms used in the industry. It presented an overview of face recognition and its application, most recent research works related to face recognition, description, and limitations about the face databases which are available to use to test the performance of these face recognition algorithms. A comprehensive summary regarding the face recognition vendor test (FRVT) 2002 made a scale evaluation in the field of automatic face recognition technology with its conclusion and finally the researchers have presented the test results of each algorithm [9].

In 2001, Viola and Jones presented the most classical approach for detecting faces from a given image by using simple cascade AdaBoost and Haar features classifiers to achieve high efficiency and high performance. After that, even most of the researchers have paid attention to enhance the accuracy of face detection by using different features like Scale-Invariant Feature Transform (SIFT) [10], Local Binary Pattern (LBP) [11], and Histogram of Oriented Gradient (HOG) [12]. But the accuracy of all these face detection techniques will be decreased crucially in the practical application as a result of facial visual diversity increases.

Other than the cascade structure in 2008 Felzenszwalb proposed HOG based Deformable Part Model detection method which is called DPM, using SVM [13]. By using this technique they were able to achieve a higher performance than the other method when using a small number of samples that are not labeled correctly. However, the method of calculation became too complex and was mostly depending on artificially designed features.

Since it is difficult to achieve high accuracy from the other the available face detection techniques the method of detecting face by using the conventional neural network (CNN) has become the major breakthrough in the industry and had become the mainstream in recent years

Zhang et al. proposed a new framework that uses a deep convolutional neural network to detect the face alignment [14] and Yang et al. carried out research on face attribute recognition via the deep convolutional neural network [15]. Most of these recent researches paid a lot of attention to identifying the untrollable part of the face area, such as posture changes, overemphasized expressions, and facial occlusion. However, it seems very difficult to achieve a high generalization ability only by using a single detection structure. In 2019 Rong et al. presented a way to overcome these problems of existing face detection techniques by using Cascade convolutional neural networks which can capture different complicated and changing situations in the face region by using a large number of training images [16].

### 2.3 Research carried under text-based recommendation.

Recurrent neural plays an important role when it comes to recommendation systems because of its capability of learning highly complex relationships from a sequence of data.

As a result of the availability of large-scale evaluation data set recently to the standard of the recommendation, systems have been significantly improved. The available recommendation approaches were categorized into three categories such as modeling-based, pre-filtering, and post-filtering approaches by Oku et al in 2007 [17].

In 2005 Adomavicius et al presented the Pre-filtering approaches approach which can be used for the recommendation by utilizing contextual information to select data for a particular context, and then use the traditional recommendation method on the selected data to predict ratings [18]. On the other hand, the Post-filtering method will use the traditional methods to predict the ratings first then use the contextual information to adjust results. The most important feature of these two methods was separating contextual information from the process of rating estimation and it leads to unsatisfying findings.

In 2016, B. Hidasi et al investigated the possible ways of utilizing the features in the Recurrent Neural Networks-based session model by combining the techniques in session-based modeling in deep learning. The researchers also presented alternative strategies for training RNNs that suit better than the standard training methodologies [19]

## 2.4 Research carried under CNN-based fashion recommendation.

Recently there is a considerable amount of research done in the field of deep learning recommendation which will identify the user preferences correctly from the given input and recommend the items that the user would like. When it comes to image processing, neural networks play an important role because of their capability of identifying the specific features in the images. But when the features that need to be extracted are at different positions of the image, neural networks would fail when they identify these features. Training neural networks with all such images would be impossible. Therefore the convolutional neural networks have come into play since it has an immense capability of extracting features without considering their position.

The researches like [20], [21], [22]-[24] have presented convolutional neural network-based approaches for clothing recommendations. The convolutional neural networks-based approach became popular with the creation of AlexNet by Krizhevsky, et al. [24]. This can be considered as a turning point in the field of fashion recommendation.

They created a network known as AlexNet, which is considered as a turning point for neural network

Recently triplet embeddings and Siamese networks have been popular [22][23]. The main advantage of using Siamese networks is it has the capability of directly training for the problem at hand. As an example, if there is a problem that we need to address by finding the homogeneity between two different cloth items, the Siamese networks will address this kind of problem directly instead of training the network to detect the objects first and then find the similarity. The research [20] presented by Veit et al. in 2015 proposed a system that uses a Siamese CNN to find similar kinds of clothes. The method named triplet embeddings also plays an important role as they help in achieving higher accuracy by extracting better features. In 2016, Jiang et al. presented a work of combining triplet embedding with cross-domain image retrieval.

Using transfer learning also has become popular in the field of neural networks. Transfer learning can be defined as the process of using the already defined weights of a trained model to train a new model on the same kind of task. This will help us to avoid the burden of training the entire new model from scratch and this will reduce the time taken for training the entire neural network. In this paper also we will be using transfer learning to obtain more accurate results efficiently.

### 2.5 Summary

This chapter discussed the research carried out under the fashion recommendation systems. The chapter was mainly dedicated to three parts since there are few related works available in the image-based fashion recommendations system. The first part discussed the research directly related to the images based fashion recommendation systems. The second part discussed the related works that are carried out under image processing since the main aim of this research is building research that provides recommendations based on the given input images. Therefore image processing, especially image preprocessing plays a significant role in this field, and the investigation of the image processing literature is essential to figure out the most suitable image preprocessing technique. The last part of this chapter was dedicated to text-based recommendations. Before the image-based recommendation became famous, the text-based recommendation was used mainly and in order to identify the problem with text-based recommendations, the related works of literature on that were investigated. And finally, we have added some literature available on CNN-based fashion recommendation systems.

# CHAPTER 3

# METHODOLOGY

As mentioned in the introduction, the intent of this research is to propose an Intelligent Cloth Recommendation System that provides a better user experience with high accuracy by addressing the problems in existing research solutions. This system will leverage the gap between the research industry and actual customer needs in the field of cloth recommendations.

The main algorithm contained in this proposed system is responsible for image-based recommendation and it takes the image of a fashion item as a main input to the system and recommends the matching clothing items by establishing the fashion embedding among the available fashion items and the input query image.

Fashion embedding is the process of embedding or translating low-dimensional and continuous vector representation of discrete variables of fashion items into dimensional vectors. The proposed solution uses a deep neural network to perform fashion embedding and the output of the fashion embedding will be used to find out the similarities between fashion items to perform the recommendations.

Deep neural networks can be defined as one of the perfect tools to map an image to a vector by ignoring the noise or the unimportant details of the input image. This proposed solution classifies the given image into the categorical labels available in the clothing domain by using deep neural networks and calculating the cosine similarity of the available images. The architecture of the deep neural networks that have been used to implement the proposed solution and the steps that were carried out to achieve a higher accuracy of the recommendation decisions can be further illustrated as follows.

The proposed image-based recommendation will consist of three main steps. As the first step, it will ask the user to upload an image based on the predefined criteria, and then it will send it to the image processing unit. The image processing unit is responsible for applying image processing techniques such as Gaussian blur on the top of the uploaded images and resizing of the images then these images are used as the input of the deep neural network available in the system to use fashion embedding to calculate the similarity among the images.

### 3.1 Data Set

This project is using the combination of two different datasets one is the DeepFashion dataset that was collected by Chinese Hong Kong University and Fashion-MNIST.

DeepFashion dataset has more than 800,000 diverse images and as well as it consists of well-defined annotations along with some additional information about the categories and landmarks. This data itself contains different types of prediction subsets which are tailored by the researchers for 5 different specific tasks. From these available subsets, one subset named Attribute Prediction can be used for the categorization and clothing attribute prediction. It consists of 290,000 images belonging to 50 categories and 1,000 clothing attributes. Mainly the dataset consists of 6 clothing categories and in order to have an even distribution of clothing items in the final dataset, in this research 3,000 images from each category will be used.

The Fashion-MNIST dataset is having 60,000 images (50,000 images of the training set and 10,000 images of testing data set) in total which are classified into 10 categories including ‘dresses’, ‘t-shirts’, ‘ trousers’, ‘pullovers’, ‘coat’, ‘sandals’, ‘bags’, ‘ankle boots’, ‘sneakers’ and ‘shirts’ [25].

The reason for using the combination Fashion-MNIST data set and DeepFashion data set is to overcome the problem of missing categories in both data sets. As an example, the DeepFashion data set would be having only the images which are related to the clothing items therefore the Fashion-MNIST data set was used to get the images of other fashion items like ‘footwear’. Combing both DeepFashion and Fashion-MNIST, and conducting a google search to find out the different types of fashion images it was possible to collect a data set of different fashion categories like ‘Dresses’, ‘T-shirts’, ‘Jeans’, ‘Shirts’, ‘Skirts’, ‘Track Pants’, ‘Tops’, ‘Trousers’, ‘Kurtas’, ‘Casual shoes’, ‘Sport shoes’, ‘Heels’, ‘Sandals’, ‘Handbags’, ‘Watches’.

These categories can be made into a hierarchy of categories by combining their common features as follows. For each of these categories, 3,000 were collected and a total of 45,000 images. From the collected images randomly selected 9000 were used for the validation and others were used for the training with the 80:20 ratio [26][27].

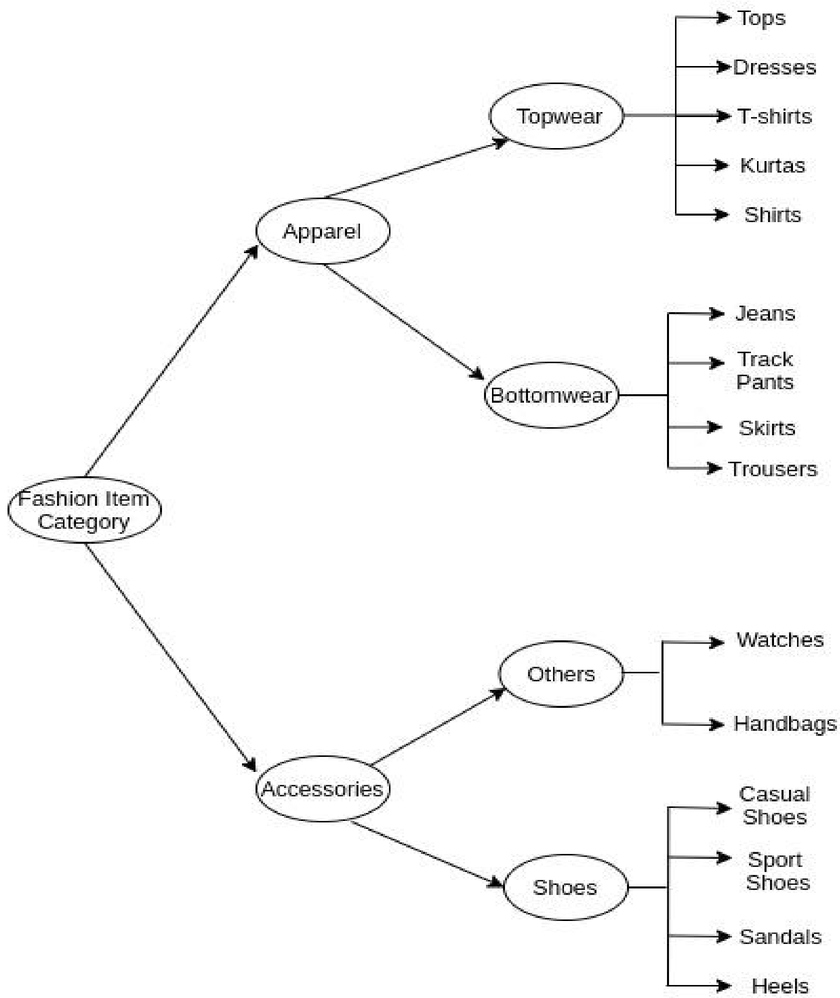



*Figure 3.1:* Distribution of Fashion Item Categories

### 3.2 Image Pre-Processing

The image-preprocessing unit was responsible for applying image preprocessing techniques on top of the collected data set and resizing the images before using them for the training or validation of the deep neural network. This was used to detect the outliers in the data set, to remove unwanted noisy data, and as well as to identify the missing values in the image and treat them accordingly to reduce their effect on the final output [28].

Image preprocessing algorithms used in this system will consist of three main steps as Read, Resize and Remove noise. Reading is the process of storing the path to the image in a variable that will help to load the image to the memory and refer to that location as a variable reference. These also will be preliminary steps for the resizing and noise removal steps since after this step, image can be accessed via the variable. Resize steps contain the implementation to define a base size for all the images. With the resizing step all the images are resized into the 64 * 64 dimensions. The final step of the image pre-processing is noise removal. In order to remove the unwanted details and the noise of the image Gaussian blurring or Gaussian smoothing is used as the image processing technique and this results in a blurred image by using the Gaussian function [29].

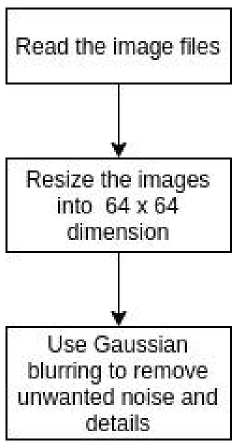



*Figure 3.2:* Steps of Image Preprocessing

### 3.3 Recommendation using CNN.

When it comes to the field of neural networks, Convolutional Neural Networks are widely used for image processing, categorization, and detection because of their ability to differentiate images from other images correctly. A Convolutional Neural Network or ConvNet/CNN is one of the available Deep Learning algorithms which can take an image as the input, assign importance in the means of learnable weights or biases to various aspects of the objects. This will provide the ability of CNN to differentiate one image from the other with higher accuracy. The pre-processing required by the Convolutional Neural Networks is much lower as compared to other algorithms available in Deep Learning [30].

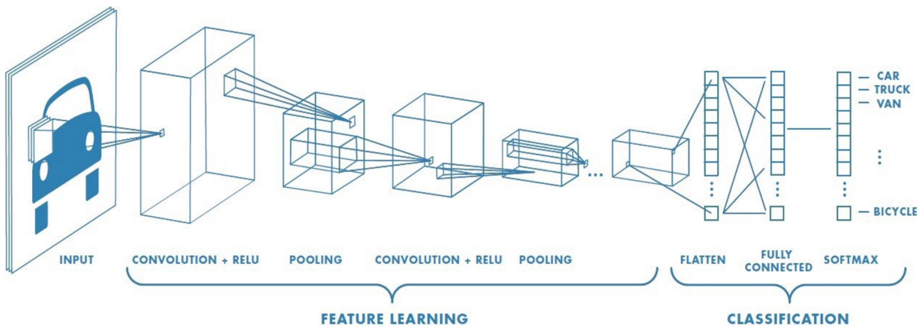



*Figure 3.3:* Architecture of Convolutional Neural Network

Rather than using CNN based approach for feature extraction and recommendation, researchers have followed out some other ways of doing image-based recommendation called "Low-level features extraction". Since it was difficult to achieve high accuracy with this it was decided to go for the CNN-based approach. In Low-level features, the extraction technique cosine similarity of the test image was calculated with the arbitrarily selected leaders of predefined clusters. This has been carried out for the following feature descriptors and finally to select the topmost recommendation the mean of the cosine similarity of these feature descriptors was calculated.

- Image texture: A most important feature which indicates the pixel distribution of the input image.
- Histogram of oriented gradients: Used as a feature descriptor to detect the objects in the given query.
- HSV histogram: This feature mainly deals with the color distribution of the image between 0 and 255 pixels. The output after obtaining the HSV histogram will be a feature vector of 255 x1 dimensions which basically represents the pixel frequency distribution of the image.
- Edge detection feature: Sobel edge detection algorithm is used for detecting the edges of the given image and the output is a feature vector of the image size.

```
import cv2
import numpy as np
from skimage.feature import hog
from skimage.io import imread
from skimage import exposure
from skimage import filters


# loading the input image
img = imread('image1.jpg')

# Obtaining Edge Detection features
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
sobel_img = filters.sobel(gray_img)

# Obtaining HOG features
fd, hog_image = hog(img, orientations=9, pixels_per_cell=(8, 8),
                    cells_per_block=(2, 2), visualize=True, multichannel=True)
hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 10))

# Obtaining Texture features
g_kernel = cv2.getGaborKernel((21, 21), 8.0, np.pi/4, 10.0, 0.5, 0, ktype=cv2.CV_32F)
texture_img = cv2.filter2D(gray_img, cv2.CV_8UC3, g_kernel)
```

*Figure 3.4:* Implementation of "Low-level features extraction"

Since it was difficult to achieve high accuracy with the "Low-level features extraction", we have decided to look for a more efficient way and accurate way of extracting more descriptive feature descriptors which can be used to compute the similarity and to return the topmost K recommendations to the user as output. As a result of that, deep learning has been selected because of its powerful capability of detecting patterns and features from images.

In order to select the best CNN to use for this proposed solution and trained that neural network on the selected data set, as the first step, the five most popular CNN based architectures such as VGG, Resnet, Inception network, MobileNet, and DenseNet were evaluated and checked how they will behave on the selected data set of this research. The features extracted from these neural networks were given as the input to the cosine similarity calculation model and then based on the results of this similarity calculation the top K scored images were output as the recommendation results.

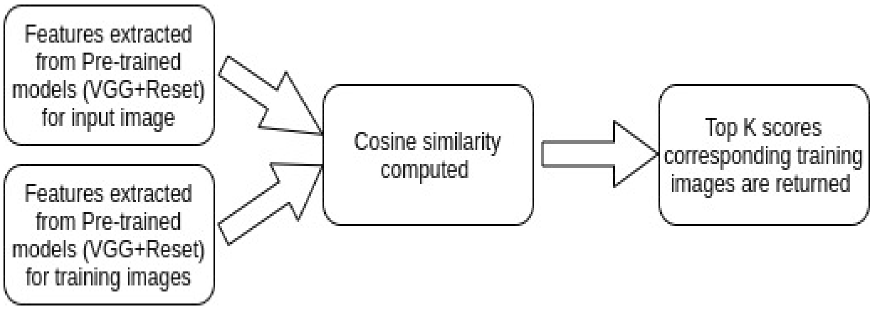



*Figure 3.5:* Components of the proposed system

When the pre-trained neural networks are used as the feature extractors, first it was required to remove the final output layer since the model has not been used for the classification purpose and only the feature extraction will be carried out for the other steps in the proposed solution. Therefore the final output layer has been removed from the model and with that step, the output of the neutral networks becomes a linear feature vector for an image that was pooled and reduced using a global average pooling layer followed by a flatten layer. Then the features vector is produced for all training images available in the data set and cosine similarity is calculated and based on the top K scores of the cosine similarity, the corresponding top K images are returned as the final recommendation output of the system.

By using precision and recall as the base model for this evaluation, as mentioned earlier five popular CNNs were evaluated over the 9000 images, and with that, it was possible to obtain that the VGG and Resnet are the best performing model for this data set. Therefore the research was continued by using VGG and Resnet for the feature extraction over the five most popular conventional neural networks.

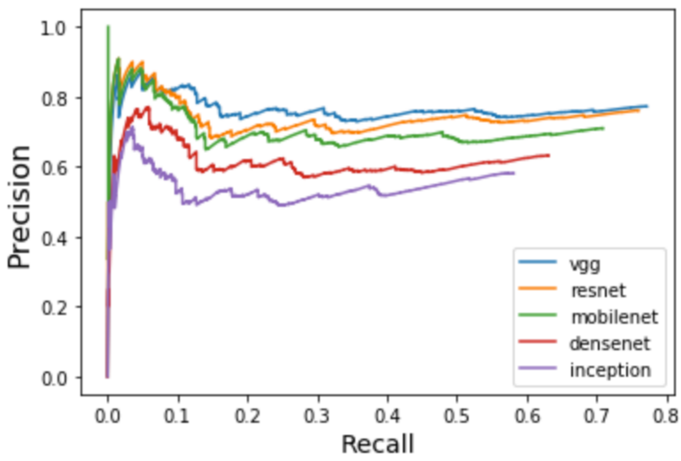



*Figure 3.6:* Performance of different CNNs

```
#defining Resnet and Vgg pre-trained Models
def resNetModel(height, width):
    model = ResNet50(weights='imagenet', include_top=False, input_shape = (height, width, 3))
    model.trainable = False
    output = GlobalMaxPooling2D()(model.outputs)
    model = Model(inputs=model.inputs, outputs=output)
    model.summary()
    return model

def vggModel(height, width):
    model = VGG16(weights='imagenet', include_top=False, input_shape = (height, width, 3))
    model.trainable = False
    output = GlobalMaxPooling2D()(model.outputs)
    model = Model(inputs=model.inputs, outputs=output)
    model.summary()
    return model
```

*Figure 3.7:* Implementation of the using pre-trained models

Rather than using either VGG or Resnet alone for the feature extraction, we have decided to use a combination of both VGG and Resnet to evaluate the results. In order to combine the results from both neural networks, the weighted average is used and since the Resnet is producing a bigger vector than the size of VGG's feature vector, SKlearn's SelectK is used as the feature reduction technique to reduce the size of the feature vector of Resnet. SKlearn's SelectK is a powerful technique that selects top K features from a set of given features based on the target value for the features. The final weighted feature representation which is obtained after applying the weighted average for both the ResNet and VGG is used to compute cosine similarity and then the top K recommendations were selected based on the top K highest scores.

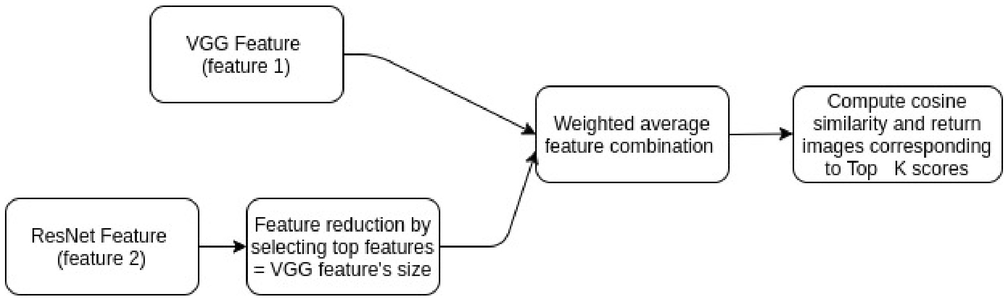



*Figure 3.8:* Components of ResNet +VGG System

After building the combination model, it was evaluated by changing the weight ratio. The model with the best performing weight ratio was selected and then it was evaluated again with the ResNet and VGG to conclude whether the combination model will give better results. Results implied that the combination model is performing with higher accuracy than the ResNet and VGG on the selected data set. Therefore, rather than using the ResNet or VGG alone, researchers have come up with a final decision to continue the research with the combination model and implement the proposed solution.

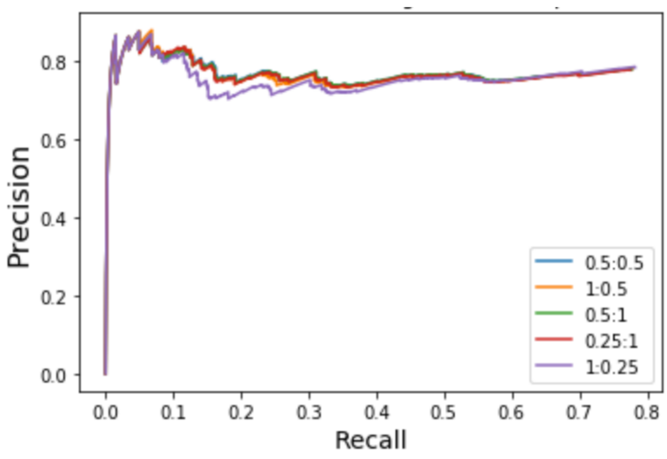



*Figure 3.9:* Results after changing the weight ratio

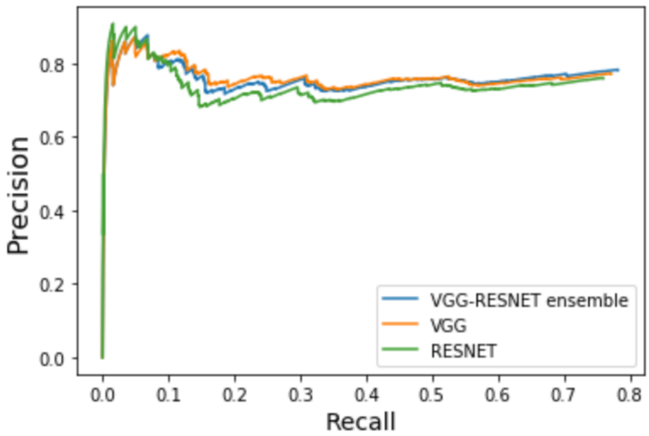



*Figure 3.10:* Results of the combination model

After deciding to choose the combination of ResNet and VGG as the final CNN to proceed with this research over the other CNNs on which we conducted the evaluation, the following steps have been carried out to train ResNet and VGG for the selected data without being overfitting and for implementing the final solution.

**3.4 Training ResNet + VGG**

When it comes to the training of CNNs two main methods are possible to follow. One is transfer learning and the other one is fine-tuning. Transfer learning involves using a pre-trained model on top of the new data set to resolve a problem that is very similar to the domain that it is trained for. But here rather than using a pre-trained model as it is, it is possible to re-trained only some of the interesting layers and freeze all the other layers. With that approach, it will give the possibility of taking the advantages of a pre-trained model and avoid the burden of training the whole model from the beginning. Here we expect to use the pre-trained CNN model as a feature extractor by removing the final layer of both ResNet and VGG, which is responsible for classifications. Therefore the necessary actions have been taken to train the final layer by freezing all the other layers in the CNN model. Usually, in a CNN model, the first layers are responsible for extracting the basic features and the final layers would be dedicated to learning the more advanced features. Therefore it would be important to train the last layers of the neural network rather than the first layers. Fine-tuning involves the steps of adjusting weights of the neural networks to get a better output on the given dataset.

Following steps have been taken over the 45,000 images to train the pre-trained ResNet + VGG to achieve better accuracy.

Steps 1: Build the base model using available pre-trained models

```
# We build the base model
base_model = VGG16(weights='imagenet', include_top=False, input_shape=input_shape)
base_model.summary()
```

Steps 2: We freeze every layer in our base model

```
# Apply transfer learning by freezing all the layers.
# This will keep the feature extractor as it is without over-fitting to the given data set
for layer in base_model.layers:
  layer.trainable = False
  print('Layer ' + layer.name + ' frozen.')
```

Step 3: Add the last layer of the pre-trained model to our classifier which calculates the Cosine similarity.

```
last = base_model.layers[-1].output
x = Flatten()(last)
x = Dense(1000, activation='relu', name='fc1')(x)
x = Dropout(0.3)(x)
x = Dense(10, activation='softmax', name='predictions')(x)
model = Model(base_model.input, x)
```

Step 4: Compile the model and start training

```
# compile the model
model.compile(optimizer=Adam(lr=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
# start training
epochs = 10
batch_size = 256
model.fit(X_train_resized, Y_train,
          batch_size=batch_size,
          validation_data=(X_test_resized, Y_test),
          epochs=epochs)
```

Step 5: Evaluate the accuracy and the loss in the test set

```
# Evaluate the accuracy and the loss in the test set
scores = model.evaluate(X_test_resized, Y_test, verbose=1)
print('Test loss:', scores[0])
print('Test accuracy:', scores[1])
```

These steps were carried out for both VGG16 and ResNet models and training was done until they meet the stopping criteria.

### 3.5 Summary

The methodology chapter provides the approach to how the proposed fashion recommendation system was designed and implemented. It provides more on-site on the data selection, how the selected data set was preprocessed and what are methodologies carried out on selecting the best CNN, and final decisions made on the decision of the system architecture.

# CHAPTER 4

# SYSTEM ARCHITECTURE AND IMPLEMENTATION

As discussed in the previous methodology, the proposed system consists of a combination of two ResNet and VGG to achieve better accuracy when extracting the features from the images and then the cosine similarity was calculated to get the top K images which return as the final recommendation for the given image.

## 4.1 ResNet Architecture.

In the proposed solution, ResNet50 has been used for the implementation. ResNet50 is the most popular variant of the available ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. RestNet50 can be introduced as one of the representative deep convolutional neural networks which are encoding of classification and auto-encoding. ResNet has been able to win the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by using feature transmission capability to prevent gradient vanishing, because of this ResNet can be trained more effectively than a much deeper network that used [31]. A subset of the ImageNet database has been used to train the RestNet50 model and it has the capability of classifying the images into 1000 object categories such as a mouse, keyboard, pencil, and many more. As explained in the previous chapter we have frozen all the layers other than the output layer of the neural network and therefore the training process will not update the weights of the frozen layers. This method of freezing the weights of many existing initial layers will significantly reduce the training time and then it will speed up the network training process and prevent the over-fitting on the new fashion dataset.

The architecture of the ResNet50 mainly consists of four main stages as shown in the following figure and the network has the capability of taking input images having width and height as multiples of 3 and 32 channel width. Each and every ResNet will perform the initial convolution and max-pooling using 7×7 and 3×3 kernel sizes respectively [31]. After this step, stage 1 of the network will be getting started and it consists of 3 Residual blocks which contain 3 layers each. The kernel sizes, which are used to perform the convolution operation in all of these 3 layers of the block of stage 1 can be denoted as 64, 64, and 128. The curved arrows denoted in Figure 12 refers to the identity connection in the neural networks and the dashed connected arrow of Figure 12 implies that the convolution operation in the Residual Block is performed with stride 2, because of that the size of the given input will be reduced to half in terms of width and height. But on the other hand, the width of the channel will be enhanced to double. That means when the input image propagates from one stage to another stage the image size would be reduced to half of its actual size and the channel width would be increased to double its actual size.

As the final layer, the network is having an Average Pooling layer which is followed by a fully connected layer, includes 1000 neurons.

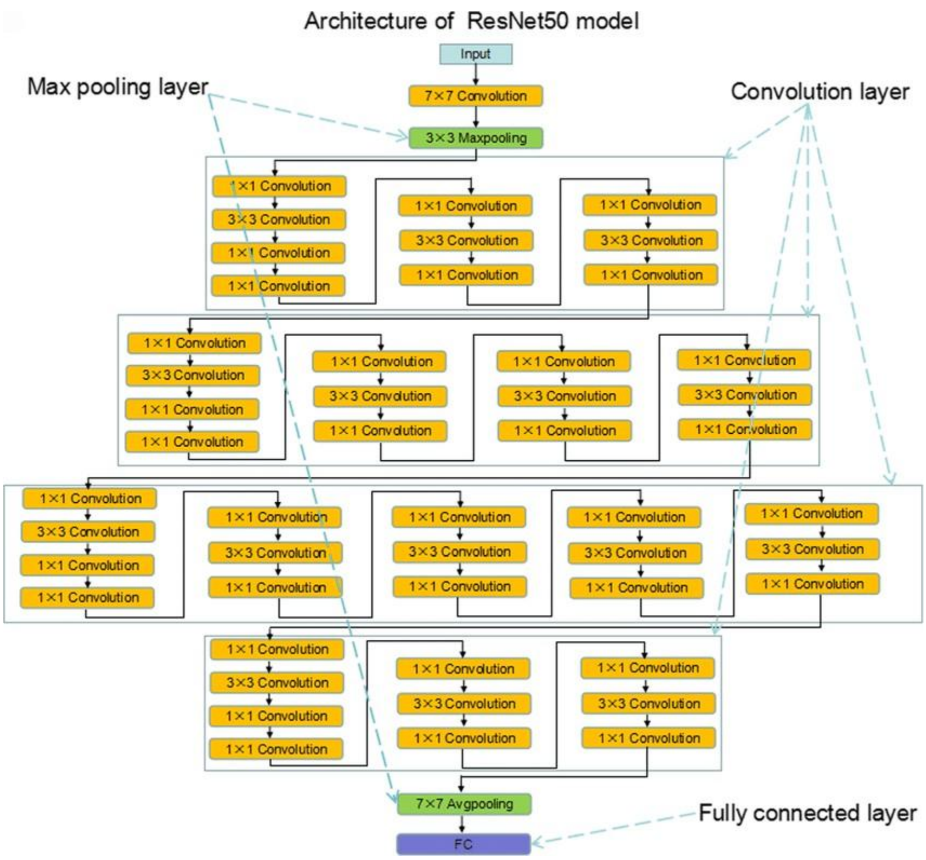
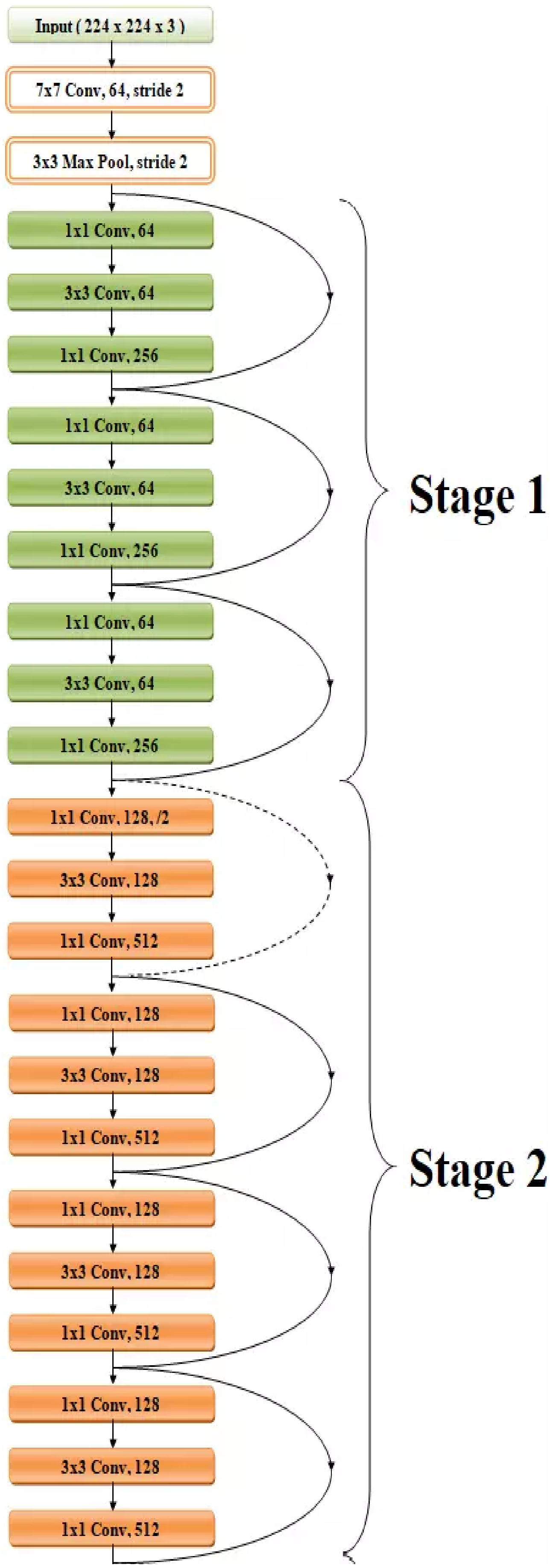



*Figure 4.1:* ResNet 50 Architecture


Input ( 224 x 224 x 3 )
7x7 Conv, 64, stride 2
3x3 Max Pool, stride 2
1x1 Conv, 64
3x3 Conv, 64
1x1 Conv, 256
1x1 Conv, 64
3x3 Conv, 64
1x1 Conv, 256
1x1 Conv, 64
3x3 Conv, 64
1x1 Conv, 256
1x1 Conv, 128, /2
3x3 Conv, 128
1x1 Conv, 512
1x1 Conv, 128
3x3 Conv, 128
1x1 Conv, 512
1x1 Conv, 128
3x3 Conv, 128
1x1 Conv, 512
1x1 Conv, 128
3x3 Conv, 128
1x1 Conv, 512
Stage 1
Stage 2

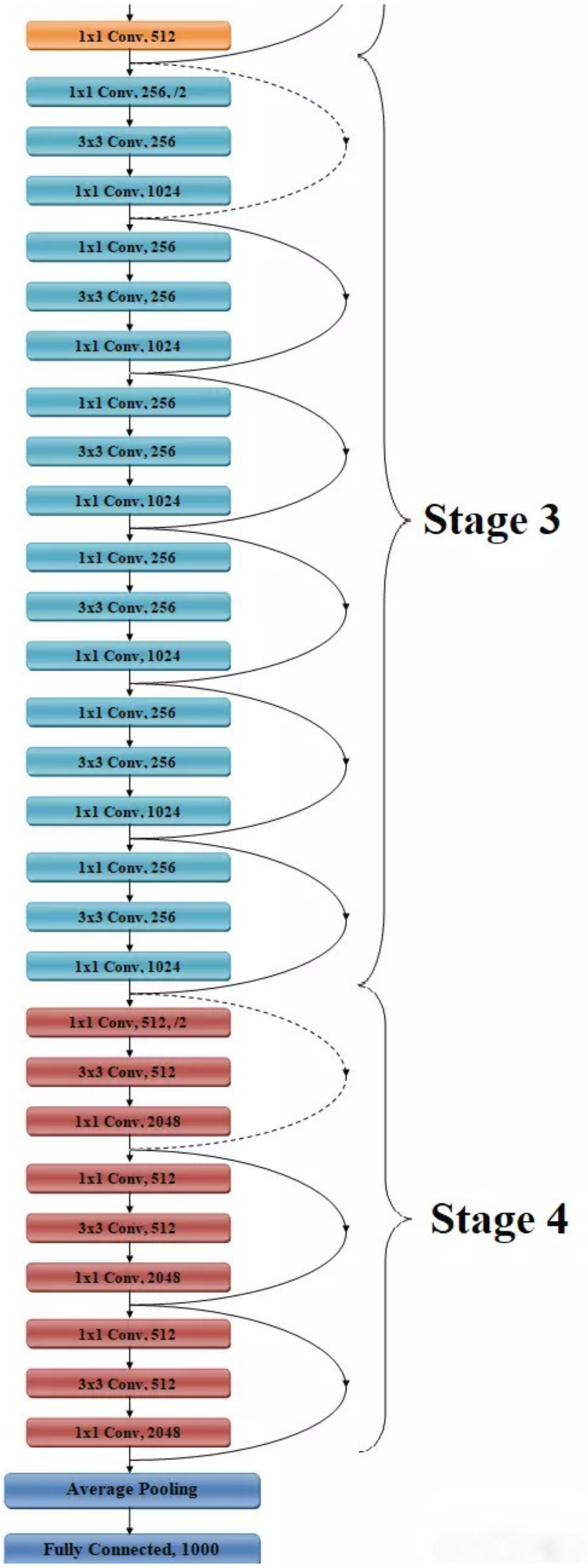



*Figure 4.2:* Layers of ResNet50

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| conv2_x | 56×56 | 3×3 max pool, stride 2 | | | | |
| | | $\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$ | $\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$ |
| conv3_x | 28×28 | $\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$ | $\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$ | $\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$ | $\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$ | $\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$ |
| conv4_x | 14×14 | $\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$ | $\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$ | $\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$ | $\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$ | $\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$ |
| conv5_x | 7×7 | $\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$ | $\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

*Figure 4.3:* ResNet Architecture

### 4.1.1 Implementation of ResNet

In the ResNet implementation, there are two types of blocks that will be used based on whether the input dimension of the network and the output dimension are the same or different [33].

- The identity block: Used in the scenario where the input activation has the same dimension as the output activation.
- The Convolutional block: This type of block is used when the input and activation and the output activation do not have the same dimension. This will be the opposite case of the identity block. The difference between these two blocks is that the Convolutional block is having a CONV2D layer in the shortcut path.

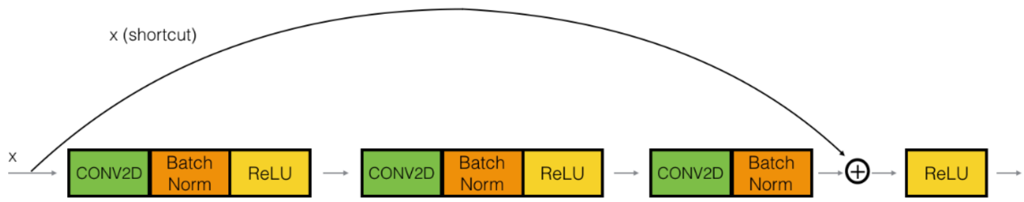



*Figure 4.4:* Identity block Structure

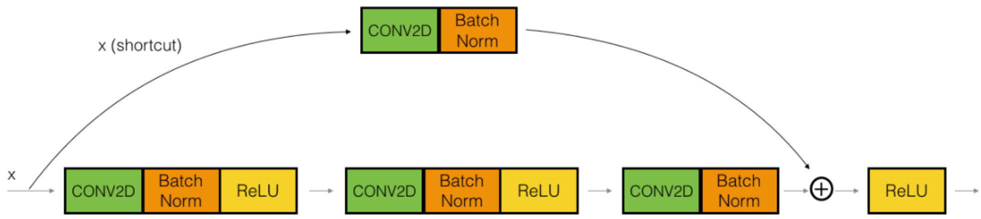



*Figure 4.5:* Convolutional block Structure

## Implementation of identity block

```
def identity_block(X, f, filters, stage, block):
    """
    Implementation of the identity block
    """

    # defining name basis
    conv_name_base = 'res' + str(stage) + block + '_branch'
    bn_name_base = 'bn' + str(stage) + block + '_branch'

    # Retrieve Filters
    F1, F2, F3 = filters

    # Save the input value. You'll need this later to add back to the main path.
    X_shortcut = X

    # First component of main path
    X = Conv2D(filters=F1, kernel_size=(1, 1), strides=(1, 1), padding='valid', name=conv_name_base + '2a',
               kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=3, name=bn_name_base + '2a')(X)
    X = Activation('relu')(X)

    # Second component of main path
    X = Conv2D(filters=F2, kernel_size=(f, f), strides=(1, 1), padding='same', name=conv_name_base + '2b',
               kernel_initializer=glorot_uniform(seed=0))(X)
```

```
    X = BatchNormalization(axis=3, name=bn_name_base + '2b')(X)
    X = Activation('relu')(X)

    # Third component of main path
    X = Conv2D(filters=F3, kernel_size=(1, 1), strides=(1, 1), padding='valid', name=conv_name_base + '2c',
               kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=3, name=bn_name_base + '2c')(X)

    # Final step: Add shortcut value to main path, and pass it through a RELU activation
    X = Add()([X, X_shortcut])
    X = Activation('relu')(X)

    return X
```

# Implementation of Convolutional block

```
def convolutional_block(X, f, filters, stage, block, s=2):
    # defining name basis
    conv_name_base = 'res' + str(stage) + block + '_branch'
    bn_name_base = 'bn' + str(stage) + block + '_branch'

    # Retrieve Filters
    F1, F2, F3 = filters

    # Save the input value
    X_shortcut = X

    # First component of main path
    X = Conv2D(F1, (1, 1), strides=(s, s), name=conv_name_base + '2a', kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=3, name=bn_name_base + '2a')(X)
    X = Activation('relu')(X)

    # Second component of main path
    X = Conv2D(F2, (f, f), strides=(1, 1), padding='same', name=conv_name_base + '2b',
               kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=3, name=bn_name_base + '2b')(X)
    X = Activation('relu')(X)
```

```
    # Second component of main path
    X = Conv2D(F2, (f, f), strides=(1, 1), padding='same', name=conv_name_base + '2b',
               kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=3, name=bn_name_base + '2b')(X)
    X = Activation('relu')(X)

    # Third component of main path
    X = Conv2D(F3, (1, 1), strides=(1, 1), padding='valid', name=conv_name_base + '2c',
               kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=3, name=bn_name_base + '2c')(X)

    X_shortcut = Conv2D(F3, (1, 1), strides=(s, s), padding='valid', name=conv_name_base + '1',
                        kernel_initializer=glorot_uniform(seed=0))(X_shortcut)
    X_shortcut = BatchNormalization(axis=3, name=bn_name_base + '1')(X_shortcut)

    # Final step: Add shortcut value to main path, and pass it through a RELU activation
    X = Add()([X, X_shortcut])
    X = Activation('relu')(X)

    return X
```

# Implementation of ResNet

```
def ResNet50(input_shape=(64, 64, 3), classes=15):
    # Define the input as a tensor with shape input_shape
    X_input = Input(input_shape)

    # Zero-Padding
    X = ZeroPadding2D((3, 3))(X_input)

    # Stage 1
    X = Conv2D(64, (7, 7), strides=(2, 2), name='conv1', kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=3, name='bn_conv1')(X)
    X = Activation('relu')(X)
    X = MaxPooling2D((3, 3), strides=(2, 2))(X)

    # Stage 2
    X = convolutional_block(X, f=3, filters=[64, 64, 256], stage=2, block='a', s=1)
    X = identity_block(X, 3, [64, 64, 256], stage=2, block='b')
    X = identity_block(X, 3, [64, 64, 256], stage=2, block='c')

    # Stage 3
    X = convolutional_block(X, f=3, filters=[128, 128, 512], stage=3, block='a', s=2)
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='b')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='c')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='d')
```

```
    # Stage 4
    X = convolutional_block(X, f=3, filters=[256, 256, 1024], stage=4, block='a', s=2)
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='b')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='c')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='d')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='e')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='f')

    # Stage 5
    X = convolutional_block(X, f=3, filters=[512, 512, 2048], stage=5, block='a', s=2)
    X = identity_block(X, 3, [512, 512, 2048], stage=5, block='b')
    X = identity_block(X, 3, [512, 512, 2048], stage=5, block='c')

    # AVGPOOL . Use "X = AveragePooling2D(...)(X)"
    X = AveragePooling2D()(X)

    # output layer
    X = Flatten()(X)
    X = Dense(classes, activation='softmax', name='fc' + str(classes), kernel_initializer=glorot_uniform(seed=0))(X)

    # Create model
    model = Model(inputs=X_input, outputs=X, name='ResNet50')

    return model
```

## 4.2 VGG Architecture

In 2014, Simonyan and Zisserman introduced VGG architecture as a very Deep Convolutional Network which can be used for a wide range of image recognition. The architecture of the VGG network consists of a conv1 layer that takes RGB images of fixed size as the input to the neural network. As well, the VGG network consists of a preprocessing layer that takes an RGB image that contains pixel values in the range of 0–255 and subtracts the mean image values that are calculated over the entire training data set [34]. These input images will be passed through a set of different convolutional layers along with the filters that have been used in these layers with a very small receptive field [35]. In VGG 16 architecture, there are 13 no of convolutional layers along with 3 fully connected layers. Instead of having large filters, VGG is having small filters (3*3) with more depth. After the convolutional layers, a pooling layer was used in the networks with a max-pool of 2*2 size and stride 2 this will have an effect on reducing the height and width of a volume from 224*224*64 to 112*112*64 [36][37]. There are two more 2 more convolution layers, followed by this layer which has 128 filters, and this will output the new dimension of 112*112*128. After the pooling layer, the size of the volume will be decreased to 56*56*128. There are two more convolution layers with 256 filters where each layer followed by a downsampling layer that will reduce the size to 28*28*256. After following the final pooling layer, 7*7*512 volume becomes flattered into a Fully Connected (FC) layer with 4096 channels and softmax output with 1000 classes. All the hidden layers available in VGG use a ReLu-activation function but only the last FC-layer uses a softmax-activation function.

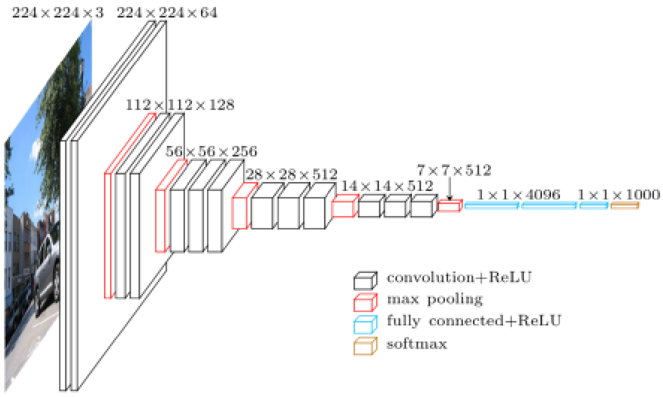



*Figure 4.6:* Visualization of VGG16 architecture

Source: [34]

### 4.2.1 Implementation of VGG

```
input = Input(shape =(224,224,3))
# 1st Conv Block

x = Conv2D (filters =64, kernel_size =3, padding ='same', activation='relu')(input)
x = Conv2D (filters =64, kernel_size =3, padding ='same', activation='relu')(x)
x = MaxPool2D(pool_size =2, strides =2, padding ='same')(x)
# 2nd Conv Block

x = Conv2D (filters =128, kernel_size =3, padding ='same', activation='relu')(x)
x = Conv2D (filters =128, kernel_size =3, padding ='same', activation='relu')(x)
x = MaxPool2D(pool_size =2, strides =2, padding ='same')(x)
# 3rd Conv block

x = Conv2D (filters =256, kernel_size =3, padding ='same', activation='relu')(x)
x = Conv2D (filters =256, kernel_size =3, padding ='same', activation='relu')(x)
x = Conv2D (filters =256, kernel_size =3, padding ='same', activation='relu')(x)
x = MaxPool2D(pool_size =2, strides =2, padding ='same')(x)
# 4th Conv block

x = Conv2D (filters =512, kernel_size =3, padding ='same', activation='relu')(x)
x = Conv2D (filters =512, kernel_size =3, padding ='same', activation='relu')(x)
x = Conv2D (filters =512, kernel_size =3, padding ='same', activation='relu')(x)
x = MaxPool2D(pool_size =2, strides =2, padding ='same')(x)
```

```
# 5th Conv block

x = Conv2D (filters =512, kernel_size =3, padding ='same', activation='relu')(x)
x = Conv2D (filters =512, kernel_size =3, padding ='same', activation='relu')(x)
x = Conv2D (filters =512, kernel_size =3, padding ='same', activation='relu')(x)
x = MaxPool2D(pool_size =2, strides =2, padding ='same')(x)
# Fully connected layers

x = Flatten()(x)
x = Dense(units = 4096, activation ='relu')(x)
x = Dense(units = 4096, activation ='relu')(x)
output = Dense(units = 1000, activation ='softmax')(x)
# creating the model

model = Model (inputs=input, outputs =output)
model.summary()
```

The proposed system architecture consists mainly of a Restnet and VGG as described in the previous chapter since that combination performed better on the selected data set than the other combination of CNNs. The output of both VGG and ResNet is combined together by using a weighted average method and then it is used as the input to the cosine similarity calculator in order to generate the top k number of recommended images as the final output of the system.

### 4.3 Recommendation using cosine similarity

For the recommendation step of this proposed system, we used the last fully connected layer of both VGG and Resnet as feature vectors. For any given images in the dataset, there will be one corresponding feature vector generated by the weighted average combination of ResNet and VGG feature vectors and it will be the input to the recommendation system which calculates the similarity between the feature vectors.

Similarity calculation will be done as the final step of this system using different measures to calculate similarity scores between feature vectors of the target image and feature vectors of all images in the target data set. In order to calculate the similarity between given pairs of images, we have tried different types of similarity calculation methods such as L2 distance, cosine distance, and neural network models.

As an example for two different images i and j, the L2 distance score can be defined as following [38][39]

$$S_{L_2} = || V_i - V_j ||_2 \qquad (1)$$

where vi, vj ∈ Rl are the two corresponding feature vectors, and l = 4096 is the length of feature vectors. The smaller the score L2 is, the more similar the two images are.

The cosine distance score is defined as

$$S_{cosine} = \frac{V_i V_j}{|| V_i |||| V_j ||} \qquad (2)$$

The larger the cosine score denotes the more similarity between given images. Rather than using other similarity calculation methods, the proposed system uses the cosine similarity calculation as the final similarity calculator since it provides more accurate results than the other methods by measuring the similarity between two non-zero vectors of an inner product space.

After calculating the similarity among the input image feature vector and the feature vector of the other available images in the target data set the top k images are identified as the similar images that are most similar to the target image.

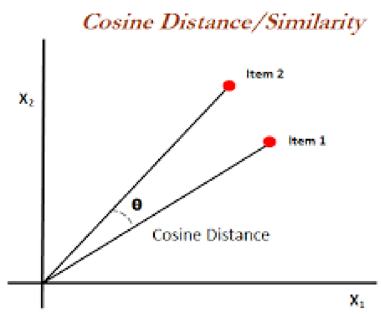



*Figure 4.7:* Cosine Similarity Calculation

Source: [40]

**Calculate cosine similarity and provide the recommendations**

```
# Calculate distance Matrix
cosine_sim = 1-pairwise_distances(df_embs, metric='cosine')
cosine_sim[:4, :4]


# Function that output the fashion recommendations based on the cosine similarity score.
def get_recommender(idx, df, top_n=5):
    sim_idx = indices[idx]
    sim_scores = list(enumerate(cosine_sim[sim_idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:top_n + 1]
    idx_rec = [i[0] for i in sim_scores]
    idx_sim = [i[1] for i in sim_scores]

    return indices.iloc[idx_rec].index, idx_sim


get_recommender(2993, df, top_n=5)
```

### 4.4 Summary

This chapter elaborates more details on the overall system architecture along with the ResNet and VGG architecture and as well as it provides the implementation details on how the final system Been implemented with the selected technologies.

# CHAPTER 5

# EVALUATION

The aim of the evaluation is to access the recommendation results produced by the proposed system. The proposed system consists of two main CNNs (ResNet and VGG) which were selected by comparing the performance of available CNN on the selected data set as mentioned in the methodology chapter. Therefore the main purpose of this chapter would be to evaluate the final output of the system by comparing the final output that was obtained by using the ResNet and VGG alone for the feature extraction. The results of this evaluation will be used to verify the validity of the design decision taken to combine ResNet and VGG together in the designing phase.

The evaluation of the proposed implementation was carried out under several matrices.

In order to measure the validity of the obtained results following matrices were used over the 20 users. Since the final aim of this proposed solution is to predict the user preferences based on the given input images two main matrices have been used to assess the performance and to determine whether the actual users like the given recommendations. Here the randomly selected 5 different query images were given to a user and asked to mark whether he or she would like the given recommendation then if they liked the recommendation output by the system it was considered as True-positive.

Recall: This was used to represent the percentage of the items that were given as the recommendation by the systems also favored by the target users. Ru is recommendations that are output by the system. Tu is all items favored by the target user [41].

$$Recall = \frac{\sum_{u}|R_u \cap T_u|}{\sum_{u}|T_u|} \quad (3)$$

Precision: Precision denotes the percentage of items that are recommended by the system and also favored by the target user among all items. Ru is recommendations that are predicated by algorithms. Tu is all items favored by the target user [41].

$$Precision = \frac{\sum_{u} |R_u \cap T_u|}{\sum_{u} |R_u|} \quad (4)$$

Then we can calculate the F1 score by combining both Recall and Precision in order to compare the system with the system along with the recommendation systems which use the VGG and ResNet alone. Table 01 summarized the results of the evaluation step. Models with the larger F1 score which is near to 1 would be considered as better models than the model with smaller F1 scores [41].

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \quad (5)$$

Table 5.1: Performance of different CNNs

| CNN | Recall | Precision | F1 score |
|---|---|---|---|
| ResNet50 | 0.85 | 0.77 | 0.81 |
| VGG16 | 0.85 | 0.78 | 0.81 |
| ResNet50 + VGG16 | 0.84 | 0.79 | 0.82 |

Other than measuring the performance of the overall system by using the above matrices, the following outputs were able to process by giving different query images. Based on the obtained output it seems that the system can detect the features of the cloth images like the clothing material, shape, and patterns in the material and especially the category with fair accuracy.

Query Image

Recommendation Results

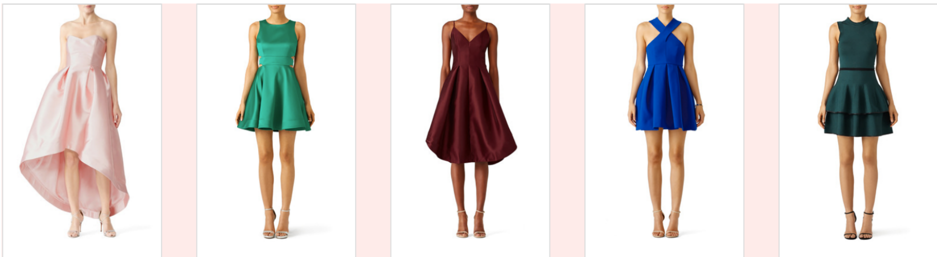
Query Image

Recommendation Results

Query Image

Recommendation Results

Query Image



Recommendation Results

*Figure 5.1:* Recommendation results obtained from the system

# CHAPTER 6

# CONCLUSION

Developing a fashion recommendation system with higher accuracy has been an emerging research area for the past few years, with the development of online marketing. Rather than stepping to the physical store nowadays people always try to purchase items in online stores. Best recommendation systems always play an important role in online stores since they save the time of the buyers and increase the sellings. Therefore most of the sellers always try to engage the best recommendation system with their websites. Most of the currently available recommendation systems focus on text-based recommendations and try to output the recommended items based on collaborative or content-based filtering. As well as even though most of the available online stores provide a text-based searching facility, image-based searching would be really uncommon in the websites. Therefore as a solution to overcome this research gap this paper proposed a system for the image-based recommendation for fashion recommendations by utilizing the feature extraction capability of deep learning.

### 6.1 Research Contribution

In this paper, we have investigated how to utilize the deep learning technique for developing an efficient and accurate fashion item recommendation System. Although there are numerous available ways to represent the compatibility of the fashion items, this problem needs to be well handled to obtain better accuracy and user satisfaction. This paper presented several convolutional neural networks available in deep learning that can be used to develop image-based fashion recommendation systems by comparing their accuracy and performance in the selected data set.

This research has been conducted by creating a rich data set of over 45000 images by collecting the relevant images from various available resources. This data is having different types of fashion images that belong to more than 15 categories. As well as, it is having a fair distribution over these categories and the future researchers will be able to use this new data set for their research or extend this data set further.

Other than that the research presented the most suitable image processing technique for preprocessing fashion images to enhance the feature extraction and propose a new way of conducting fashion recommendations by applying a combination of the most powerful convolutional neural networks in deep learning (ResNet and VGG). This methodology can be used for the development of an image-based recommendation system in the future with different technologies and since this research presented the evaluation, researchers would be able to save their time in the future when they select a CNN for their fashion recommender. The proposed solution contains steps for training the neural networks, how to use CNN as a feature extractor than the traditional classifier and the best weighted averaged parameters for combining the feature vectors from ResNet and VGG.

### 6.2 Limitation of the research approach

The current dataset used for the implementation and testing of this proposed system will only contain around predefined fashion categories like 'Dresses', 'T-shirts', 'Jeans', 'Shirts', 'Skirts', 'Track Pants', 'Tops', 'Trousers', 'Kurtas', 'Casual shoes', 'Sport shoes', 'Heels', 'Sandals', 'Handbags', 'Watches'. Therefore the current system will show higher accuracy when recommending the images belonging to these categories and will have a somewhat lesser accuracy when we use an image belonging to other categories.

Based on the evaluation of the current model the maximum F1 score that was possible to achieve was 0.82 this score is higher than the other available CNN's but would be able to achieve a better accuracy even higher than this if we could retrain some other layers of the ResNet and VGG. But because of the resource limitation, it was not possible to train the layers other than the output layers.

Most of the available recommendation systems will take collaborative or content-based filtering as the filtering mechanism for the recommendation system; this proposed system does not have the collaborative or content-based filtering capability, but it can be added as an improvement.

### 6.3 Future Works

For the present implementation, pre-trained ResNet and VGG were used by freezing most of the layers. It would be possible to achieve higher accuracy by training some of the earlier layers of both of these neural networks which extract the basic features. Feature works of this research would include identifying the most important layers of the ResNet and VGG in terms of features extraction of the images and will follow the necessary steps to train those layers using transfer learning and fine-tuning.

In order to achieve better performance, this system can be further extended by using the output of the last layer of the ResNet and VGG as feature representation, there is a possibility of using the model to classify the image first and then calculate cosine similarity only among the members of the classified class. This will help to speed up the calculations by avoiding irrelevant calculations.

As well as, it is possible to evaluate the possibility of using Generative adversarial neural networks for this system for getting accurate recommendations.

## REFERENCES

[1] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in ACM SIGMOD Record, 2000, vol. 29, pp. 1-12.

[2] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in Proc. 25nd Annu. Int. SIGIR Conf., 2002, pp. 253-260.

[3] Zhang H, Ji Y, Li J, et al. “A Triple Wing Harmonium Model for Movie Recommendation,” IEEE Transactions on Industrial Informatics, 2016, 12(1): 231-239.

[4] T. Iwata, S. Wanatabe, and H. Sawada, “Fashion coordinates recommender system using photographs from fashion maga-zines,” IJCAI, 2011, 22(3): 2262.

[5] A. Veit, B. Kovacs, S. Bell, et al., “Learning visual clothing style with heterogeneous dyadic co-occurrence,” in Proc. IEEE Int. Conf. Comput. Vision, pp. 4642-4650, 2015.

[6] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, “Image-based recommendations on styles and substitutes,” in Proc. 38nd Annu. Int. SIGIR Conf., pp. 253-260, Aug. 2015.

[7] Liu, Si, Luoqi Liu, and Shuicheng Yan. "Magic mirror: An intelligent fashion recommendation system." In 2013 2nd IAPR Asian Conference on Pattern Recognition, pp. 11-15. IEEE, 2013.

[8] Y. Li and L. Cao and J. Zhu and J. Luo, Mining Fashion Outfit Composition Using an End-to-End Deep Learning Approach on Set Data, IEEE Transactions on Multimedia, vol. 19, no. 8, pp. 1946–1955, 2017

[9] Tolba, A.S., El-Baz, A.H. and El-Harby, A.A. Face recognition: A literature review. *International Journal of Signal Processing*, *2*(2), pp.88-103, 2006.

[9] Chen, Huizhong, Andrew Gallagher, and Bernd Girod. "Describing clothing by semantic attributes." In *European conference on computer vision*, pp. 609-623. Springer, Berlin, Heidelberg, 2012.

[10] Yeqing Ren, Youqiang Sun, Xuechun Jing, Zhihua Cui, and Zhentao Shi, Adaptive Makeup Transfer via Bat Algorithm[J], Mathematics, 2019, 7(3): 273, DOI: 10.3390/math7030273;

[11] B. Yang, J. Yan, Z. Lei, S. Z. Li, "Aggregate channel features for multi-view face detection", IEEE Int. Joint Conf. Biometrics, pp. 1-8, 2014.

[12] Zhihua Cui, Lei Du, Penghong Wang, Xingjuan Cai, and Wensheng Zhang. Malicious code detection based on CNNs and multi-objective algorithm[J], Journal of Parallel and Distributed Computing, 2019, 129: 50-58. DOI: 10.1016/j.jpdc.2019.03.010.

[13] P. Felzenszwalb, D. McAllester, D. Ramanan, "A Discriminatively Trained Multiscale Deformable Part Model", Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008.

[14] Z. Zhang, P. Luo, C. C. Loy, X. Tang, "Facial landmark detection by deep multi-task learning", Eur. Conf. Comput. Vis., pp. 94-108, 2014.

[15] ] S. Yang, P. Luo, C. C. Loy, X. Tang, "From facial parts responses to face detection: A deep learning approach", IEEE Int. Conf. Comput. Vis., pp. 3676-3684, 2015.

[16] Qi, Rong, Rui-Sheng Jia, Qi-Chao Mao, Hong-Mei Sun, and Ling-Qun Zuo. "Face detection method based on cascaded convolutional networks." IEEE Access 7: 110740-110748, 2019.

[17] Oku, Kenta, Shinsuke Nakajima, Jun Miyazaki, and Shunsuke Uemura. Investigation for designing a context-aware recommendation system using svm. *In Proceedings of the International MultiConference of Engineers and Computer Scientists*, IMECS 2007, pages 970–975. 2007.

[18] Adomavicius, Gediminas, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin.
Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems (TOIS), 23(1):103–145. 2005.

[19] Hidasi, B., Quadrana, M., Karatzoglou, A. and Tikk, D., September. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 241-248), 2016.

[20] J. McAuley, C. Targett, Q. Shi and A. van den Hengel, "Image-based recommendations on styles and substitutes", Proc. of the 38th Int. ACM SIGIR Conf. on Res. and Develop. in Inform. Retrieval - SIGIR '15, 2015.

[21] X. Zhang, et al., "Trip outfits advisor: location-oriented clothing recommendation", IEEE Trans. on Multimedia, pp. 1-1, 2017.

[22] S. G. Eshwar, G. G. Prabhu J, A. V. Rishikesh, A. N. Charan and V. Umadevi, "Apparel classification using convolutional neural networks", 2016 Int. Conf. on ICT in Business Industry & Government (ICTBIG), 2016.

[23] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala and S. Belongie, "Learning visual clothing style with heterogeneous dyadic co-occurrences", 2015 IEEE Int. Conf. on Comput. Vision (ICCV), 2015.

[24] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2017

[25] Xiao, H., Rasul, K. and Vollgraf, R., Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*,2017.

[26] Cuimei, Li, Qi Zhiliang, Jia Nan, and Wu Jianhua. "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers." In 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), pp. 483-487. IEEE, 2017.

[27] Wang, Shoujia, Wenhui Li, Ying Wang, Yuanyuan Jiang, Shan Jiang, and Ruilin Zhao. "An Improved Difference of Gaussian Filter in Face Recognition." *Journal of Multimedia* 7, no. 6 (2012): 429-433.

[28] Bradski, Gary, and Adrian Kaehler. "OpenCV." Dr. Dobb's journal of software tools 3 (2000).

[29] AŞIROĞLU, Batuhan, Mehmet İlkay ATALAY, Alkan BALKAYA, Erden TÜZÜNKAN, Mustafa Dağtekin, and Tolga ENSARİ. "*Smart Clothing Recommendation System with Deep Learning*." In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1-4. IEEE, 2019.

[30] Catarious Jr, David M., Alan H. Baydush, and Carey E. Floyd Jr. "Characterization of difference of Gaussian filters in the detection of mammographic regions." *Medical Physics* 33, no. 11 (2006): 4104-4114.

[31] He, K., Zhang, X., Ren, S. and Sun, J., Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778), 2016.

[32] Yamazaki, M., Kasagi, A., Tabuchi, A., Honda, T., Miwa, M., Fukumoto, N., Tabaru, T., Ike, A., and Nakashima, K., Yet another accelerated sgd: Resnet-50 training on imagenet in 74.7 seconds. arXiv preprint arXiv:1903.12650, 2019

[33] Rezende, E., Ruppert, G., Carvalho, T., Ramos, F. and De Geus, P., December. Malicious software classification using transfer learning of resnet-50 deep neural network. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 1011-1014). IEEE, 2017.

[34] Yu, W., Yang, K., Bai, Y., Xiao, T., Yao, H. and Rui, Y., June. Visualizing and comparing AlexNet and VGG using deconvolutional layers. In Proceedings of the 33 rd International Conference on Machine Learning, 2016.

[35] Mateen, M., Wen, J., Song, S. and Huang, Z, Fundus image classification using VGG-19 architecture with PCA and SVD. Symmetry, 11(1), p.1, 2019.

[36] Muth, Jennifer L., and Thomas F. Cash. "Body‐Image Attitudes: What Difference Does Gender Make? 1." *Journal of applied social psychology* 27, no. 16 (1997): 1438-1452.

[37] Chua, Leon O., and Tamas Roska. "The CNN paradigm." IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 40, no. 3 (1993): 147-156.

[38] Zhang, Haijun, Wang Huang, Linlin Liu, and Xiaofei Xu. "*Clothes collocation recommendations by compatibility learning*." In 2018 IEEE International Conference on Web Services (ICWS), pp. 179-186. IEEE, 2018.

[39] Ruschendorf, L. and Rachev, S.T., A characterization of random variables with minimum L2-distance. Journal of multivariate analysis, 32(1), pp.48-54, 1990.

[40] Rahutomo, F., Kitasuka, T. and Aritsugi, M., October. Semantic cosine similarity. In The 7th International Student Conference on Advanced Science and Technology ICAST (Vol. 4, No. 1, p. 1), 2012.

[41] Wang, R. and Li, J., July. Bayes test of precision, recall, and f1 measure for comparison of two natural language processing models. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 4135-4145), 2019.