Fashion Recommender System Using Deep Learning

Pratik Variyampattil

M.Sc. in Computing
in Big Data Analytics
and Artificial
Intelligence

2020

Computing Department, Letterkenny Institute of Technology, Port Road, Letterkenny, Co. Donegal, Ireland.

Fashion Recommender System Using Deep Learning

Author: Pratik Variyampattil

Supervised by: Karen Bailey

A thesis submitted in partial fulfilment of the requirements for the

Master of Science in Computing in Big Data Analytics and Artificial Intelligence

Submitted to Quality and Qualifications Ireland (QQI)
Dearbhú Cáilíochta agus Cáilíochtaí Éireann

August 2020

Declaration

I hereby certify that the material, which l now submit for assessment on the programs of study leading to the award of Master of Science in Computing in **Big Data Analytics and Artificial Intelligence**, is entirely my work and has not been taken from the work of others except to the extent that such work has been cited and acknowledged within the text of my own work. No portion of the work contained in this thesis has been submitted in support of an application for another degree or qualification to this or any other institution. I understand that it is my responsibility to ensure that I have adhered to LYIT's rules and regulations.

I hereby certify that the material on which I have relied on for the purpose of my assessment is not deemed as personal data under the GDPR Regulations. Personal data is any data from living people that can be identified. Any personal data used for the purpose of my assessment has been pseudonymised and the data set and identifiers are not held by LYIT. Alternatively, personal data has been anonymised in line with the Data Protection Commissioners Guidelines on Anonymisation.

I give consent for my work to be held for the purposes of education assistance to future Computing students at LYIT and it will not be shared outside the Computing Department of LYIT. I understand that my assessment may be shared with any other third party and will be held securely in LYIT in line with the Records Retention Policy.

Pratik Variyampattil

Signature of Candidate Date 31-08-2020

Acknowledgments

I wish to express my special thanks of gratitude to my thesis supervisor, "Ms. Karen Bailey" for her able guidance, encouragement and support throughout my research.

I would also like to s how my gratitude to the staff and lecturers of LYIT for all the skills and knowledge I have gained during my time here, without them the research would not have been possible.

I wish to thank my family and friends who encouraged me and made me to submit my research within the limited time frame.

Pratik Variyampattil

Abstract

Technology has evolved and has brought rapid growth and change to almost every field of technology and commerce over the past decade. The clothing industry is one of the industries which has changed rapidly. Clothing is a basic need of every human. Clothing sense or style has changed over the years and the world is moving forward to new fashion every season. Some people are interested in fashion trends, some are interested in their personal styling. Many people invest a lot on clothing. To meet such growing requirement by large groups of people, organization and companies in the clothing domain some organisations are considering adopting a model that would help in meeting the customer fashion requirements. This is where the recommendation system comes into action. The recommendation system offers the potential to play an important role in contributing to-increased sales and improved customer satisfaction and this idea is becoming state-of-art in smart online shopping on e-commerce websites.

This research focuses on a Fashion Apparel Recommendation System and the mechanism behind it. We will be mainly focusing on image recognition and classification as these two are the main aspects of a recommendation system. We will compare various algorithms under Machine learning and Deep Learning implemented under Supervised and Unsupervised Learning. Under Supervised learning the data from the image is extracted as the pixel values and passed into the algorithms for training and the output will be categorical values which are the names of the apparels. In case of unsupervised learning the images are first pre-processed with the steps which include object position identification, edge detection and normalization. Later on, image feature extraction is done with the help of algorithms and are sent for clusters. We will then compare the final results of the algorithms to decide the best model for the Fashion Apparel Recommendation System.

After observing the results and accuracies obtained by the algorithms, it was seen that in Supervised Learning 5 Layered CNN performed better and gave high accuracy in Deep Learning whereas in Machine Learning XGBoost gave the better results compared to other algorithms, but in the case of Unsupervised Learning, algorithms showed poor results because of lack of dataset.

Keywords:- Apparel Recommendation system, Fashion, Machine Learning, Deep Learning.

List of Acronyms

KNN: K-Nearest Neighbour

CNN: Convolutional Neural Networks

SVM: Support Vector Machine

VGG: Visual Geometry Group

ReLu: Rectified Linear Unit

ResNet: Residual Networks

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

MLP: Multi-Layer Perceptron

PIL: Pillow

CSV: Comma Separated Values

PCA: Principle Component Analysis

ML: Machine Learning

DL: Deep Learning

Table of Contents

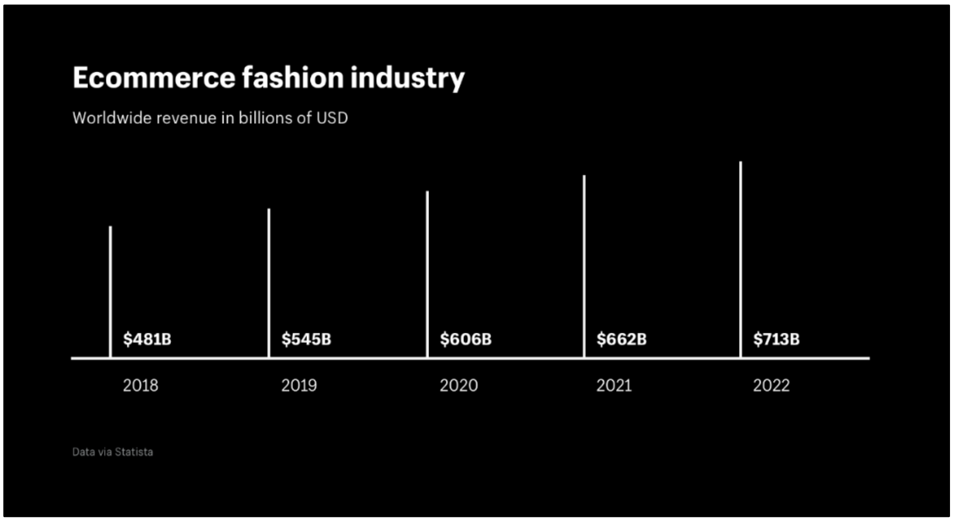
Table of Figures

Chapter 1 Introduction

1.1 Background

We have observed a lot of growth in the internet from edge connectivity to GSM and to fastest 5Ginternet facilities from the recent years.(carritech, 2017)it became open to majority of population with the advancements that happened in internet(carritech, 2017) a hughe demand was seen for many domains to take a shift from traditional marketing approaches to digital market by making their products virtual just to get maximum reach from the customer’s side(shopifyplus, 2020), sectors such as banking finance and clothing are slowly digitalizing their businesses, I am considering apparel industry for this research purpose because it has shown a drastic change in the e-commerce world, it plays a major role in contributing towards the world economy as well as it is a basic aspect of human need(shopifyplus, 2020).

If we see the statistics, the Apparel industry generated revenue in the year 2018 is close to 481 billion USD and today at 2020 it hiked from 481 to 606 billion USD which is an increase of 20% in the business over the span of only 2 years, according to the predictions by the end of 2020 the overall estimated revenue will be around 713 billion USD(shopifyplus, 2020), the below figure shows that there is a linear growth from the year 2018-2022, so we can clearly understand that there is a huge scope in this particular domain.



Worldwide fashion industry revenue from 2018-2022 adopted from *(shopifyplus, 2020)*

Also along with this growth there is a rapid development seen in the technology over this period, artificial intelligence, computer vision, machine learning and deep learning are the most popular technologies among those(Kumar and Sharma, 2016), so the fashion industry started integrating these technologies in order to understand their strength and weaknesses in their current business model and consistently kept updating their market strategies('Content based Apparel Recommendation System for Fashion Industry', 2019). The available amount of huge data is utilised by the companies to understand the customer interests, feedback and current trends with the help of these technologies(Guan, Qin and Long, 2019). The companies are adapting the recommendation engines to provide customers better online services and easy online shopping, because these recommendation engines could be easily modified for each and every customer by recommending similar products to them based on their likes, their purchase history, and their search history.('Content based Apparel Recommendation System for Fashion Industry', 2019)

Although the ecommerce industry generates a lot of revenue there are many technical challenges that needs to solved in order to give customers perfect service(shopifyplus, 2020). Storing and processing large amount of data is available in the form of images, texts and also videos are few of the challenges, the apparel images need to be labelled in order to recommend the apparels in the Apparel recommendation systems, for doing do it requires a lot of manual works so companies are trying to automate this by using machine learning and deep learning techniques and algorithms, these technologies are growing very rapidly and in coming years a lot of growth is expected and estimated(Guan *et al.*, 2016).

This particular research focuses on recommendation system and 85% of this research would be consisting of image recognition and image classification by implementing different machine learning and deep learning algorithm under both Supervised and Unsupervised learning approaches precisely images with labels and without labels to find out which learning approach is the best for recommendation purpose and which algorithms gives better results, of which the remaining 15% would be focusing on recommending the apparels.

Different aspects of studies will be discussed in further chapters, the problem statement will be discussed in chapter 2, where few question would be posed related to this research, all the literature review related contents in regard to previous work done, in the this field, technologies related to image processing, machine learning, deep learning, and differtn types of recommendation systems would be addressed in chapter 3. Chapter 4 consists of design and methodology in which overall design and flow of the research is explained and results and findings are discussed in chapter 5 and chapter 6 consists of the answers to questions addressed in chapter 2 and lastly chapter 7 consists of conclusions and future works.

1.2 Problem Context

there are many aspects which comes into picture while to provide more accurate and robust recommendations, mostly recommendations are based on the text based data like customer's cart details, their wish list, their purchase history and their interests, image dataset is another way of recommending the apparels to the customers where there is no need of textual data, this is where the research is going to focus on and more of the research would be related to computer vision, machine learning and deep learning, accuracy is the main challenge around the image-based recommendations, there are some cases where the algorithm or model predict the apparel present in the image inaccurately, and also there is a severe chance of overfitting and underfitting. Labelling process is the another issue faced by the current recommendation systems as there are millions of images available and manually labelling those images is a very difficult task and it requires a lot of time to do that, so to tackle such problems an algorithm need to chose which will overcome these challenges here in this research I am going to implement various machine learning and deep learning algorithm for image classification and image clustering, and after comparing all the results obtained by these algorithms the model which provide higher accuracy will be considered for recommendation.

Based on these challenges I am going to answer two questions which are given below:

- Can we predict the apparel class and recommend the similar apparels using images alone as an input and to what extent?

- Which Algorithm suits best for recommending the apparels using the image dataset under Machine learning and Deep Learning?

Chapter 2 Literature Survey

2.1 Evolution of Apparel industry

In today's digital era information overload is the major concern('Kumar and Sharma - 2016 - Approaches, Issues and Challenges in Recommender S.pdf', no date), data which is available to us is varied and it is present in a huge amount, because of this situation there is a severe problem of getting confused to what to choose and decide on one thing, specifically e-commerce websites generates a lots of data day by day, these data is comprised of customer's, choices about products, their personal information like their phone number, street no. and more, and also their transactions which they had made in their past. These data which is generated on daily basis could be used very efficiently by the website to boost their online business, but if we look through a customers perspective there are varied options available online, which makes it difficult to choose a particular product to solve this problem the most effective solution is recommendation system which will help customers to make right choices according to their preferences or likes('Kumar and Sharma - 2016 - Approaches, Issues and Challenges in Recommender S.pdf', no date)

But first let us take a look on the evolution of clothing industry before we dive into the recommendation system, ('2019 - Content based Apparel Recommendation System for Fa.pdf', no date)it started in 17th century in UK when industrialization and urbanization welcomed the citizens of UK to purchase the clothes according to their preferences, as a result the citizens purchasing ready made garments increased over the raw cotton textile, due to this reason markets started the supply of ready made garments from raw cotton textile, clothing industry plays a vital role to contribute towards the world economy, because of its demand fashion industry requires immense amount of supply , a steady rise is been seen in the sales of clothing by $3.3 billion annually since 2008 and globally it has exceeded $1.25 trillion in 2012, this clearly shows that the market growth for the apparel industry is rising very rapidly.

2.1.1 Technology in Apparel Industry

Advancement in technology has taken the apparel industry to the next level of sales, now it has made selling products online via e-commerce so that customers need not to step outside the house instead just place an order by just one click('2019 - Content based Apparel Recommendation System for Fa.pdf', no date), due to the online shopping there are variety of information and options to choose from, some customers find difficulties in deciding on one product out of the varied options available, so here comes the need of a Apparel recommender system that could recommend customers the products or similar products according to their likes by filtering the digitized data with the help of suitable algorithms('2019 - Content based Apparel Recommendation System for Fa.pdf', no date), A recommender system is technique or method which is used by not only the apparel industry but also by the domains such as online entertainment platforms such as Netflix and YouTube, e-commerce websites such as clothing and shopping websites, online travel websites and many more, for example a recommender in entertainment platform will recommend which movie to watch according to the user's genre preferences, implementing such a system will eventually meet customers requirement and also increase the profits of the platform using it('Kumar and Sharma - 2016 - Approaches, Issues and Challenges in Recommender S.pdf', no date)

2.1.2 Importance of fashion

Fashion in clothing plays a vital role in our everyday lifestyle either it is official or unofficial , fashion defines who we are, every organization or environment has its own set of fashion rules for example a person working in office needs a completely formal attire so that they can look professional in their respective jobs , same like that we can't go dressed up in formals for a party it needs a different set of attire to go with the environment, similarly fashion changes seasonally, we wear clothes according to our environment when it is cold we prefer sweaters, jackets, basically woollen clothes whereas in summers it changes to cotton clothes, shorts, tank-tops, bikinis, t-shirts and more among other similar clothes, so it is also important to understand fashion, fashion gives the idea of who we are, but as the trend of clothing industry (Fashion) is changing frequently there are varied options of fashion out there,(Kalra, Srivastava and Prateek, 2016) a Google research carried out in India suggests that third most searched topics are fashion related and the queries are increasing annually at the rate of 66%, so when people go online to buy the products they have multiple choices to select from, so they tends to get confused while shopping online on e-commerce websites to deal with this problem Machine Learning and deep learning techniques were implemented, traditionally text based methods were used to recommend the products to the user (Jo et al., 2020), but eventually this technique was not that satisfactory. You Only Look Once(YOLOv3) an object detection technique and Region-based Convolutional Neural Network (R-CNN)('INSTANCE SEGMENTATION GENERATOR FOR FASHION IMAGES USING DEEPFASHION-2 DATASET AND MASK R-CNN.', no date) were implemented which gave object detection in computer vision a new transformed way to detect the objects.

2.1.3 Why Recommendation system

A recommendation system is an engine that filters the information for predicting similar products to the user. Recommender system works on algorithms which help users to identify or choose the products which they like based on their likes, these systems save the time of the user and also the organization using such recommender systems remains in beneficial regarding the sales of the product, Recommender systems provide users the output based on their search history or the product which they are searching currently, this makes browsing items easier for users, the products they need from the varied amount of data present in front of them (van Meteren and van Someren, no date). Entertainment industries such as movies and songs platforms are using this kind of system more to improve the experience for users using it. Recommender Engines are of three types: Content-based filtering, Collaborative based filtering, and Hybrid (Singhal, Sinha and Pant, 2017).

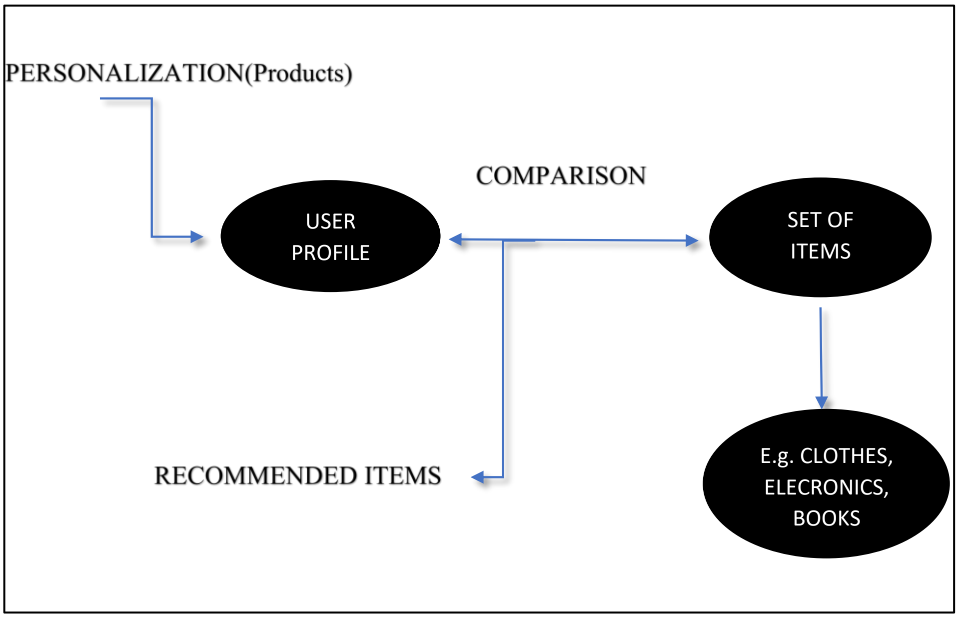


Figure 1 General model of Recommender system adapted by(van Meteren and van Someren, no date)

2.1.3.1 Content based Filtering

Content based filtering is an algorithm which recommends the products or items which are relevant for the users based on their past interactions with the products, the item representation in content based filtering is based on two approaches, which are implicit and explicit feedback, explicit feedback is based on the ratings which users gives to a particular product, and implicit feedback is mapping the amount of user spent time on viewing a particular product, whether the item was purchased or not and more(van Meteren and van Someren, no date). Similarly, a content-based system could also be developed using deep learning techniques and algorithms such as Yolo, Matrix Factorization, LSTM, LDA, Probabilistic Matrix Factorization and many more (Pratik, 2020).

2.1.3.2 Collaborative based Filtering

collaborative based filtering algorithm takes user's interdependence and recommend the user relevant sets of products based on their likes.

(Hu, Yi and Davis, 2015)Proposes a tensor factorization method which takes the user-user and item-item interactions. In the paper(Kalra, Srivastava and Prateek, 2016) they develop a deep learning recommendation system which eventually recommends the user its personalized products using "CAFFE" which is a deep learning framework used for classification of clothes and it's attributes, and Conditional Random Fields (CRF) which helps to learn the correlations of attributes of users such as the body type of the user, ethnicity, and many more.

2.1.3.3 Hybrid based Filtering

Hybrid based Filtering is the combination of content based, collaborative based and knowledge based filtering, such type of models are developed through integrating two or more types of filtering systems for a better output(Zhang et al., no date).

2.1.3.4 Deep Learning

(Kim, 2016)this book states Deep Learning as, Deep Learning is a sub field of machine learning, it has a special characteristic which is, it learns from experience on its own so when a neural network is trained several times it learns from the training period and provide suitable output, so there is no need of a manual program to embed the knowledge into a computer. (Zhang et al., no date) This survey explains many reasons why deep learning is beneficial for recommendation system compared to traditional methods or techniques, it states one of the major characteristic of a neural network is it is end-to-end differentiable which means a neural network could be designed and manipulated according to the requirement of a given problem, and (CNN) learns from its experience which is also a strong point to use a convolutional neural network.(Zhang et al., no date) this survey also discusses about the strength of neural networks and limitations of Convolutional Neural Network, it categorises the recommendation model according to the deep learning techniques.(Kang et al., 2017) This paper proposes a system where the system which recommends existing products and as well as also recommending new product images which are similar to the user preference, for this purpose Siamese-CNN framework was used and implemented. ('difference between ai and ml', no date)"Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience." As deep learning has transformed through past years it has proved to be an efficient way to solve complex problems such as object detection, feature extraction, business development and so on, there are numerous reasons to choose deep learning as a solution to a data-driven problems.

2.2 Deep Learning in the field of image processing and computer vision:

Deep learning is emerging day by day so fast that many organizations, industries and many other sectors are adopting it for optimal solutions to the problems, image processing is one of the major part of deep learning and computer vision, image processing includes, data extraction, feature extraction and so on.(‘image proccesing.pdf’, no date) image processing mainly involves the techniques of image pre-processing, segmentation and edge detection. (‘what-is-the-difference-between-computer-vision-and-image-processing’, no date) Image processing is a technique for modification of images through mathematical functions and transforming it, which means algorithms are used to modify to sharpen, smoothen, stretching, and contrasting of an image. (Eapen, 2014)this paper explains a model which is used in medical field to detect Computed Tomography-Angiography (CTA) images of the abdomen the below image shows the flow of image processing technique to reduce the noise in the images scanned, in pre-processing they implemented image resizing, cropping, histogram equalization before image segmentation

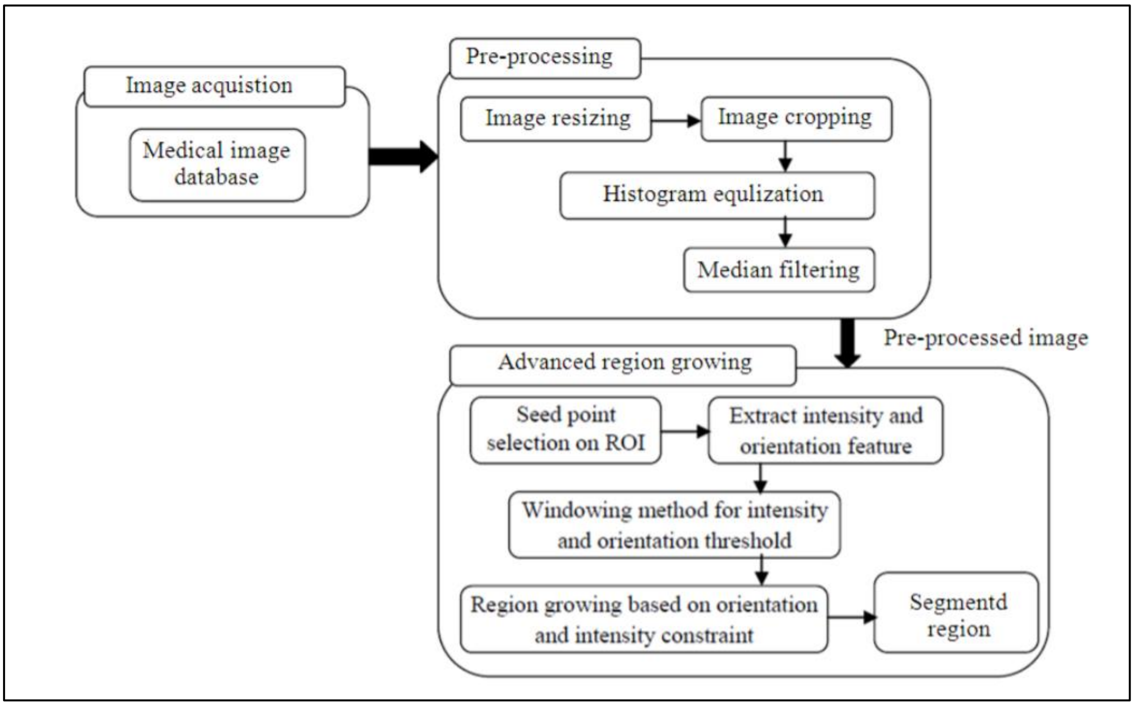


Figure 2 Flow of image processing in medical domain (Eapen, 2014)

For all these deep learning and machine learning algorithms images are required to detect features and extract meaningful features from the digitized images, so we will now look after some different types of images followed by image pre-processing techniques.

2.2.1 RGB image:

RGB stands for Red, Blue and Green, an RGB image is a combination of these 3 layers image which is basically a 3D array stored in a digitized manner, this digitized image are stored in the form of pixels which have the values either zeros or ones called as bits(GeeksforGeeks, no date), these pixels are a combination of Red, Green and Blue intensity values at a particular pixel location.(‘McREYNOLDS and Blythe - Illustration and Artistic Techniques.pdf’, no date), the value intensity of these colours i.e. Red, Green and Blue are stored in an 3 dimensional array, the range is typically in between 0-255, for instance “RGB(150,255,1)” where 150 denotes colour intensity value of Red, and 255 denotes the value of Green and 1 denotes the value of Blue, a colour image of the size 1080x1280 comprise of 13,82,400 pixels(GeeksforGeeks, no date), the quality of an image depends upon these pixels which means more amount of pixels indicates the higher quality of an image, (‘image-types.html’, no date)if we manipulate the intensity values of these colours we get different type of colours, approximately 16 million colours could be generated with the help of the intensity manipulation according to a study, below figure shows exactly how the pixels are stores in MxN sized image.

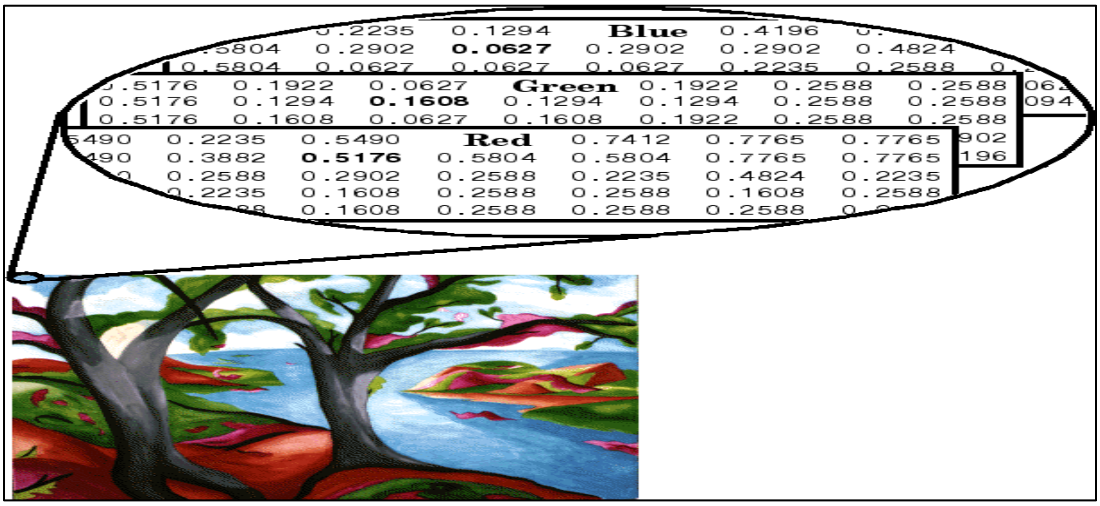


Figure 3 Pixel values of an RGB image adapted from (‘image-types.html’, no date)

As we can see the highlighted spot is an pixel of that image and it is comprised of 3 layers which is Red, Green, Blue, it has the intensity values at that highlighted spot.

2.2.2 Grayscale images:

A grayscale image is a combination of various shades of grey, when all the colours are removed from an image we get a grayscale image.(‘Jeyalaksshmi and Prasanna - 2017 - Measuring distinct regions of grayscale image usin.pdf’, no date), the pixel intensity values which are strong are considered as white and whereas the weakest pixel intensity value is considered as black(‘Jeyalaksshmi and Prasanna - 2017 - Measuring distinct regions of grayscale image usin.pdf’, no date), 8-bits can be stored in one pixel for a grayscale image, 0-255 is the range of a pixel value so 256 values of grey could be formed (‘image-types.html’, no date), considering the fact that the grayscale image stores only one channel we can conclude that the pixel stores only single intensity values in it.(‘Jeyalaksshmi and Prasanna - 2017 - Measuring distinct regions of grayscale image usin.pdf’, no date), the pixel value of an grayscale image is single intensity value meaning the pixel value 0 denotes black colour of that region and 1 denotes white color. Below figure shows how grayscale value stores the pixel values in 2-D array matrix and the pixels are stored in a single layer(‘image-types.html’, no date)

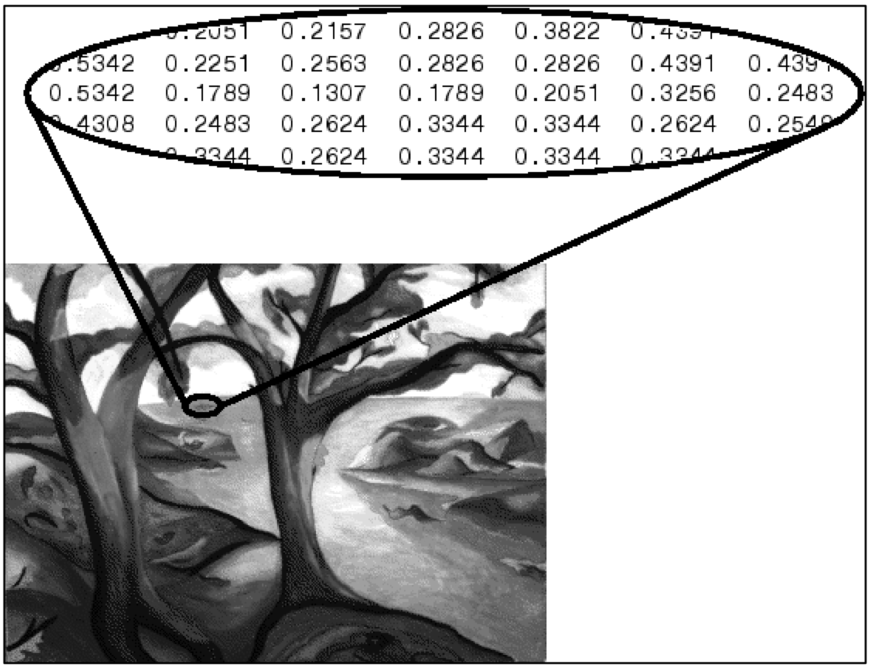


Figure 4 Figure 4 grayscale image representation adapted from (‘image-types.html’, no date)

2.2.3 Image compression

Image compression is a subfield of data compression, it is a technique in which the real image's few bits are encoded, image compression is done to reduce the storage capacity of an image and to store it in less space, due this technique the redundancy of image also get altered, basically image compression is done for the sake to modify the image in such a way that it looks exactly as the real image and also to save the image in such a way that it occupies less amount storage.('Nethaji and Shanmugasundaram - 2020 - The analysis and manipulation of a digitized image.pdf', no date) Images digitally stored into a memory while they are in their real form are always in a matrix format, it is represented as matrix A= (x, y), where X is the height of the matrix and Y is the width of the matrix, and each element in the matrix portrays the intensity value of that particular pixel.('Raju and Karthikeyani - 2012 - Improved Satellite Image Preprocessing and Segment.pdf', no date) In this paper for the purpose of image processing satellite images were taken, mainly image segmentation includes two steps that is image enhancement and image segmentation, that means image segmentation is done after some processes done on the image prior to it which is image enhancement thus image enhancement is treated as pre-processing step, in this paper image enhancement is done to correct the intensity of an image, it adjusts the contrast of an image, and it does enhancement on the edges of image and lastly left out noises are removed of the image, in this paper image enhancement is done by a proposed hybrid model, the model combined wavelets, it used CLAHE to improve the image which is input, now (CLAHE) which stands for Contrast Limited Adaptive Histogram Equalization is an special class of adaptive histogram equalization algorithm, it works like given an image it maximizes the contrast of the whole image by adapting to enhance the contrast of each pixel which are related to its closest parts, and anisotropic diffusion technique to remove the noises from image in less amount of time and maintaining the same level of quality of an image. Three algorithms were used in this paper for the purpose of segmentation process which were k-means algorithm for clustering, conventional mean-shift algorithm and modified watershed algorithm, watershed algorithm produced better segmentation results compared to other two algorithms through various experiments.

2.2.4 Edge Detection

Edge detection is a technique through which we can analyse the edges of an image just through comparing group of pixels in an image, group of pixels are compared according to its pixel intensity in a matrix and on that basis the edges of an image are calculated. ('Canny - 1986 - A Computational Approach to Edge Detection.pdf', no date) in this paper Canny proposes a computational model for edge detection, it based on the detection and localization criteria in a mathematical form, he used numerical optimization to find optimal operators for ridge edges. ('Chen et al. - 2018 - Edge detection based on machine vision applying to.pdf', no date) this paper used an edge detection based technique to cut the laminated woods, it describes the techniques based on machine vision and edge detection, an algorithm was developed which was them embedded in laminated wood edge cutting system, to detect the edges they used canny algorithm, the edged of the woods were presented by the camera calibration, computing the position for adjustments , image pre-processing and edge detection.

2.2.5 Image segmentation:

after the image processing is done next step is image segmentation, ('Hesamian et al. - 2019 - Deep Learning Techniques for Medical Image Segment.pdf', no date) Image segmentation on the medical image was done in this paper, this paper has given some most common used techniques for image segmentation and also it discusses their merits as well, they discussed about the approaches, Network training techniques and challenges, in their approaches they discussed about CNN, 2D CNN, 5D CNN,3D CNN, FCN, cascaded FCN CFCN, FOCAL FCN, Multi-stream FCN, 2D-UNET,3D-UNET, VNET, CRNs, RNN'S, LSTM, contextual CLSTM,GRU, clockwork RNN and in their network techniques they put forth some techniques which includes DEEPLY SUPERVISED, WEAKLY SUPERVISED, TRANSFER LEARNING,NETWORK STRUCTURES,ORGAN AND MODALITY, DATASETSIZE. Challenges included limited annotated data, data augmentation, transfer learning, patch-wise training, weakly supervised learning, sparse annotation, effective negative set, class imbalance, they also discussed the challenges faced in training deep learning techniques which are overfitting, training time, gradient vanishing, organ appearance, and 3D challenges.

('INSTANCE SEGMENTATION GENERATOR FOR FASHION IMAGES.pdf', no date) in this paper instance segmentation is done using Masked Region based Convolutional Neural Network (R-

CNN), this algorithm combines object detection and image segmentation for the purpose of achieving instance segmentation. Comparing the segments of objects that belongs to the same cluster or group and providing all of the instances separately is the method of instance segmentation. Below figure illustrates how and Masked R-CNN works for instance segmentation.

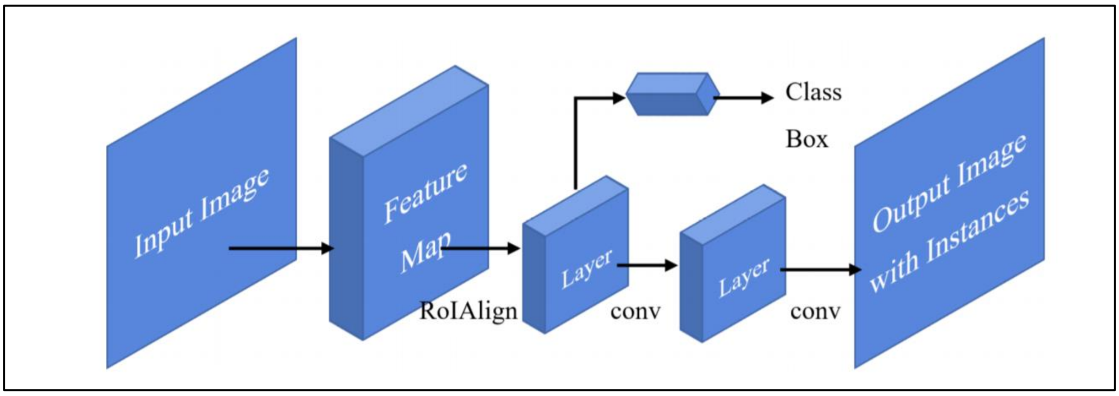


Figure 5 Figure 5 Masked R-CNN instance image segmentation

2.3 Supervised Machine Learning

Supervised Machine Learning is a type of Machine learning where the data is labelled, it is one of the techniques where the algorithm is trained based on the relationships between the input and output variable(j brownlee, 2016).supervised learning algorithms are mostly used for classification and regression problems(j brownlee, 2016), algorithms based on supervised learning approach should consists of data with labels and these labels to train the model which means that it should have categorical values for classification like "Blue or Red" and numerical values for regression based problems such as weights and sales values.(mark ryan, 2015)

2.3.1 Unsupervised Machine Learning

"Unsupervised learning is an approach of learning where instances are automatically placed into meaningful groups based on their similarity" ('Károly et al. - 2018 - Unsupervised Clustering for Deep Learning A tutor.pdf', no date) this paper has given knowledge about the unsupervised learning concepts and has noted down the surveys on clustering algorithms it has also described the advancements which took place in unsupervised learning, distributed clustering and ensembles of clustering are properly explained in this paper. (Xu and WunschII, 2005) Unsupervised learning is used when we have less amount or no knowledge base regarding the dataset , the data in these dataset are unlabelled so unsupervised machine learning is an optimal solution to discover hidden relationships between data points or to uncover some basic structures associated with the data, when those relationships or structures are revealed we get a knowledge about the dataset, unsupervised learning algorithms forms groups or clusters of datapoints considering its similarity or the distance between them. ('unsupervised machine learning webpage', 2019)this webpage article for unsupervised machine learning explains what unsupervised machine learning is, it also explains its types such as Hierarchical clustering, K-means clustering ,K-NN (k nearest neighbours),Principal Component Analysis(PCA),Singular Value Decomposition, Independent Component Analysis, it also explains the demerits of unsupervised machine learning as well as its application.

2.4 Machine Learning and Deep Learning Algorithms:

2.4.1 Convolutional Neural Networks (CNN):

Since the late 1980s Convolutional Neural Network (CNNs) had been used for performing various tasks related to computer vision, but still having a fewer applications of CNN, ('Rawat and Wang - 2017 - Deep Convolutional Neural Networks for Image Class.pdf', no date)it was barely used till the mid-2000s after that due to the improved development in algorithms and huge amount of generation of labelled data the year 2012 has seen a drastic growth of Neural Networks, this paper also discusses about the CNN used in the field of image classification and also they discuss about the existing methods of CNN and up to recent state-of-art, this paper has the information regarding CNN from its birth, development , and existing applications and algorithms.

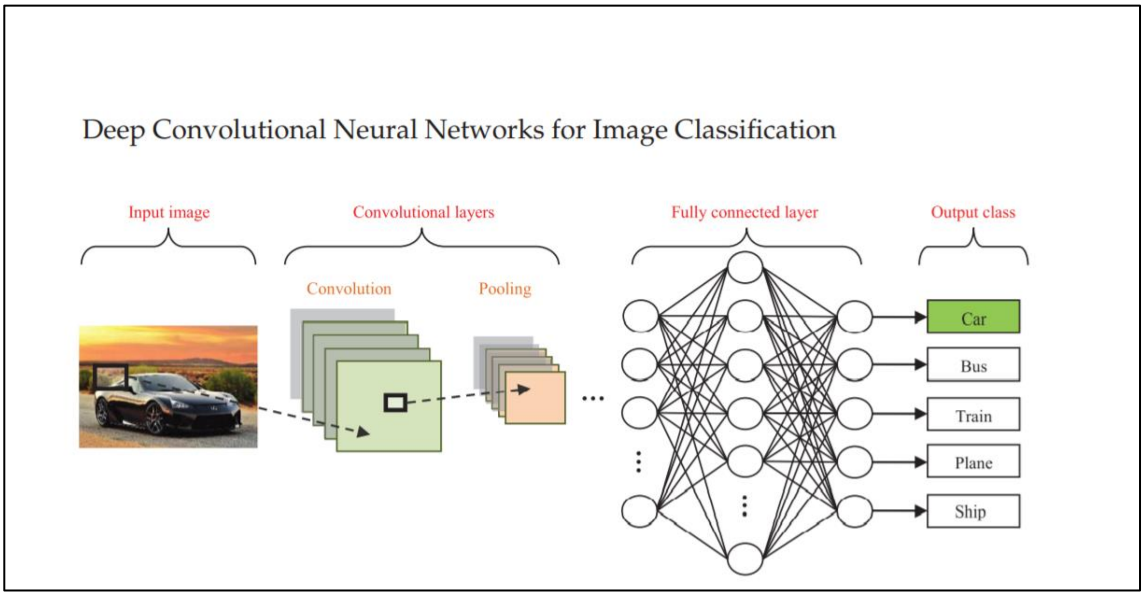


Figure 6 Architecture of Deep CNN for image classification (Rawat and Wang, 2017)

Convolutional Neural network are also called as feedforward networks because the data which is given as input in CNN takes place in such a manner that its just one direction like from input to output. Convolutional Neural Networks (CNNs)are similar as the Artificial Neural Networks (ANNs), because CNNs are also biologically inspired from the human brain, which has a combination of layers of simple and complex cells, generally CNN consists of Convolutional layers which is further a combination of convolutional and pooling layers, figure 2 illustrates CNN for an toy car classification and it clearly portrays how generally a CNN works and output is given although there modifications happening in case of improving the accuracy and reducing the cost for computation, main task of convolutional layer is to extract features from the input image, so when an image is given a input it learns the feature representations in an image and this features are further passed on to feature maps which consists of neurons in convolutional layers, and these neurons have receptive fields, and these neurons are connected to neighbour neurons in previous layers which has a set of trainable weights, pooling layers in CNN helps to reduce the spatial resolutions used by feature maps and eventually it helps to achieve invariance to input distortions and translations.('Simard et al. - 2003 - Best practices for convolutional neural networks a.pdf', no date) this paper proposes a CNN technique for document analysis, they used MNIST dataset for their experiment and achieved great result in the end, this paper proposed a technique combining CNN and elastic distortion of data which significantly improved the performance, this paper also discussed about the issues which were related to the training set size and Convolutional Neural

Networks. (‘understanding CNN’, no date) this webpage article has illustrate the flow and working of an Convolutional Neural Network, the CNN algorithm used in this article is divided into parts which is feature learning and classification, below figure illustrates the same.

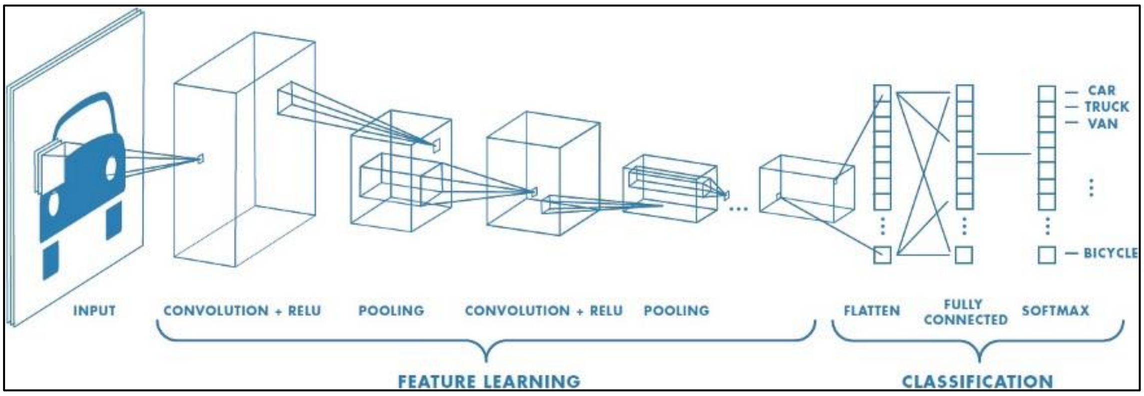


Figure 7 Understanding the flow of CNN from(‘understanding CNN’, no date)

2.5 VGG16

Visual Geometric Group or the VGG is a group from the Oxford University, this group proposed a network which is VGG16, this network is named VGG16 after looking at its 16 layers of trainable parameters, (‘the-architecture-and-implementation-of-vgg-16-b050e5a5920b’, no date) it also has a Max pool layers but without trainable parameters, a series of convolutional neural network was released by the VGG research group starting from VGG11-VGG19, the motivation behind this release of networks was just to observe the changes occurring in accuracy of the models of image classification and recognition due to the depth of convolutional networks.

2.5.1 VGG16 Architecture

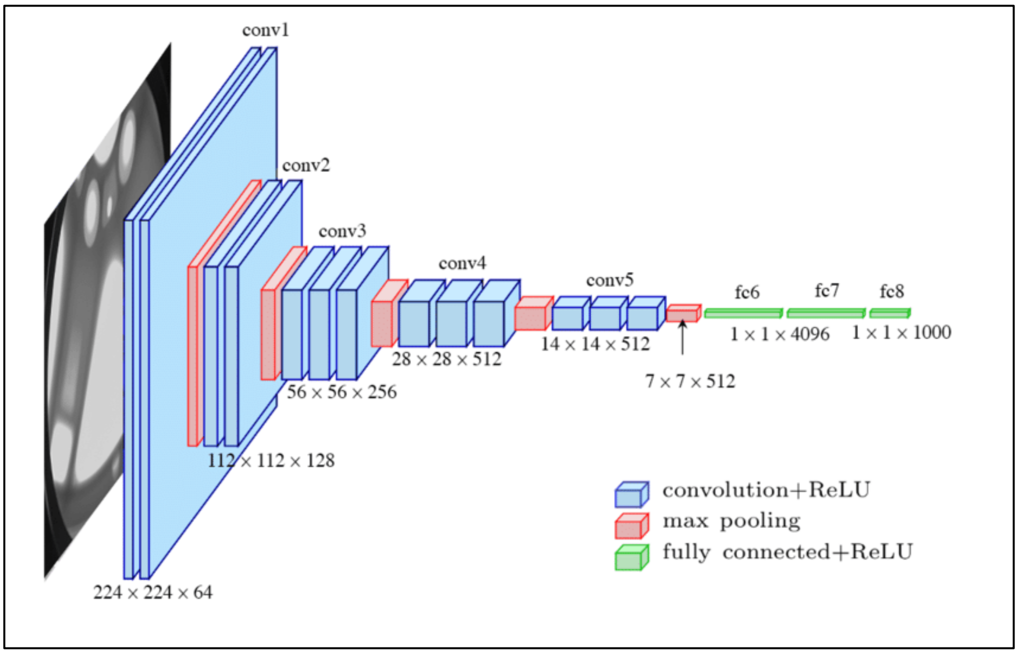


Figure 8 Architecture of VGG16 adopted from ('the-architecture-and-implementation-of-vgg-16-b050e5a5920b', no date)

The above image is an VGG16 architecture the blue rectangles in the image denotes convolutional neural layers side by side non-linear activation fucntion with it, the activation function here is rectified Linear unit commonly known as Relu, in this image there are exactly 13 blue rectangles which means it has 13 convolutional layers and 5 red rectangles which denotes 5 max-pooling layers, the green rectangles at the end represents the fully connected layers. When we do some basic math here we come to know why this network is termed VGG16 because after calculation we come to know that there are 16 layers of tuneable parameters of which 3 is fully connected layers and 13 is convolutional layers and in the output we have soft max layer which has 1000 outputs per image category. This paper also explains the features of VGG16.

2.6 ResNet50

Resnet is a network which is very powerful and has achieved several awards for its performance, ('understand-implement-resnets', no date)it had achieved for classification challenge in the ILSVRC 2015 , this network won the first place on ImageNet Localization, ImageNet Detection, COCO segmentation and detection in ILSVRC and COCO 2015 competitions, Resnet is similar to other networks as it also contains pooling layers, convolution, activation and fully connected layers, there is only one difference that makes this network different and that is identity connection between the layers the below image shows how the identity connection is going from input and sinking to the end of the residual block.

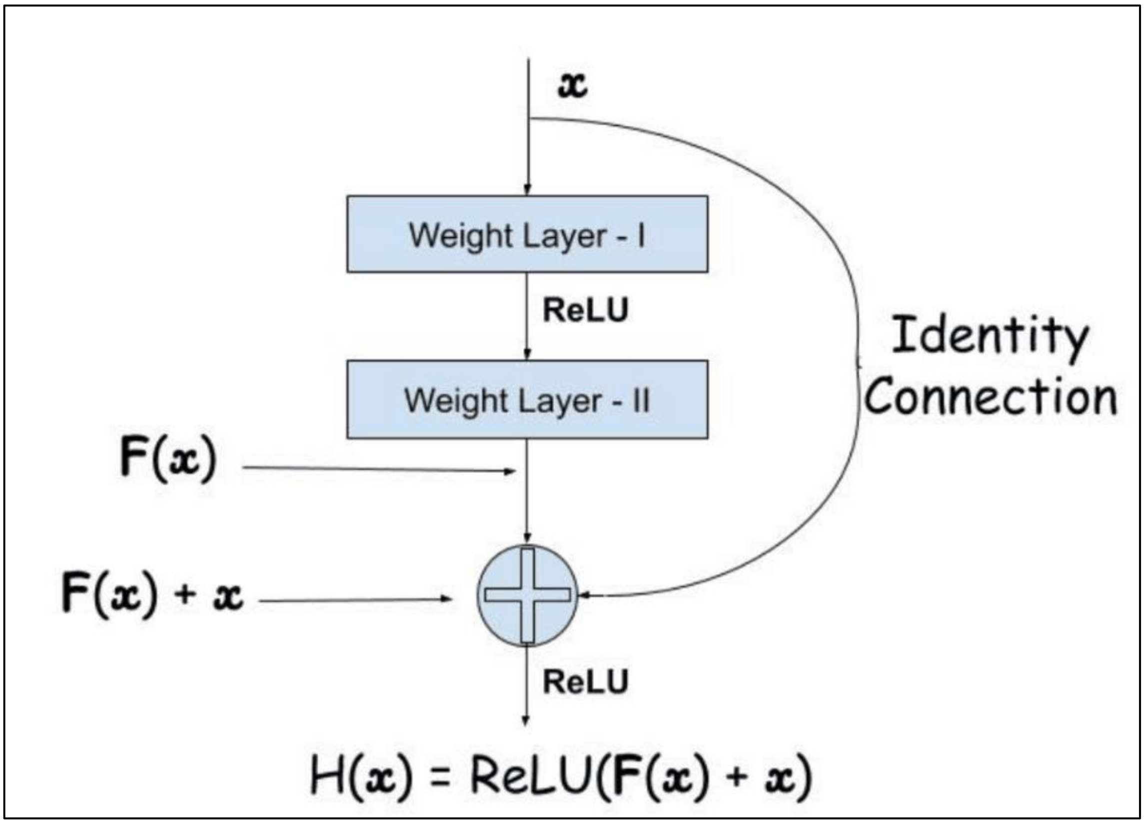


Figure 9 Resnet function adapted from('understand-implement-resnets', no date)

('deep-learning-using-transfer-learning-python-code-for-resnet50-8acdfb3a2d38', no date) Residual Networks is a network built with multiple neural layers it is a pretrained transfer learning algorithm, a network will learn more about an image if that network is embedded with more and more layers but when the layers increase at some point in time the model will eventually loose its performance because of the term called vanishing gradient problem, the figure below denotes the vanishing gradient problem.

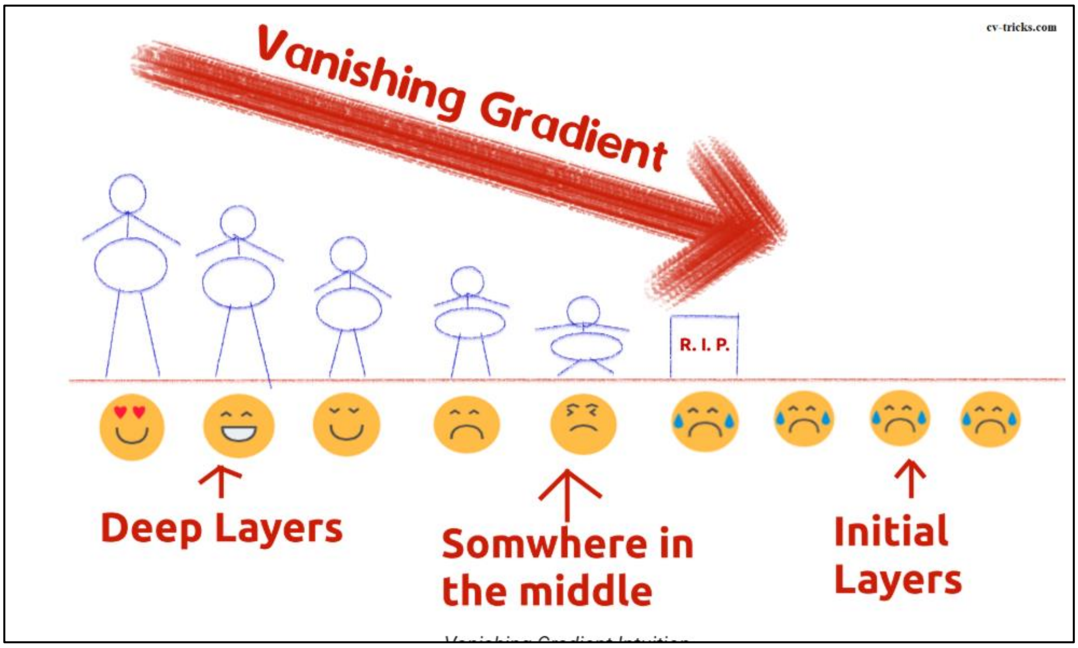


Figure 10 Gradient descent adopted from ('understand-implement-resnets', no date)

2.7 Support Vector Machine(SVM):

(KD nuggets, no date)Support Vector Machine is an algorithm which belongs to supervised machine learning algorithm, it is mostly used to solve regression and classification problems, the method of SVM algorithm is, it performs several data transformations which are very complex and separates the data on the basis of data labels, the objective of this algorithm is that it identifies the optimal boundary between the outputs, this boundary is also termed as the hyperplane, by performing like this it probably segregates the data points and most probably the data classes as well, SVM also supports non-linear data which means it is not necessary that the hyperplane should be a straight line, this helps in understanding more complex relationships between data.

2.8 K-NN Algorithm:

K-NN algorithm or K-Nearest Neighbour algorithm is an machine learning algorithm which deals with classification and clustering problems('machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761', no date), if the problem is based on the classification then this algorithm requires output variables for training the model but if it is a clustering problem then it doesn't require any output variables. This algorithm identifies the similarities between the data points and trains the model, and the clusters and classes are calculated by the Euclidean distance between these data points, KNN algorithm identifies the data points which are similar in nature and identifies them as one class, there are many applications of KNN algorithm one of them is customer segmentation, customer segmentation is used by the organizations to segment their customers based on their purchase history and also for identifying the different types of customer groups such as most recent customers, premium customers, regular customers, old customers and so on.('machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761', no date), when there are so many values of categories there is a need to decide upon the clusters basically which is "K" to give input to the KNN algorithm, deciding the clusters is an important task, in most of the times the organization using it decides the number of clusters but there are cases when we are not sure about deciding the cluster at that time we use statistical approach to decide upon these clusters(Robert GOVe, 2017).

2.8.1 Deciding the number of clusters:

Deciding the K value for KNN algorithm is very important task and to decide upon that clusters Elbow Method is used (Robert GOVe, 2017), the number of clusters are identified with the help of this method associated with a graph where the x-axis denotes the number of clusters and y-axis denotes the sum of square errors, so basically what happens is the dataset is passed to the K-means algorithm for a range of numbers and suppose 20, so this algorithm runs 20 times on the dataset and calculates the sum of square errors for each iteration it does, after doing so the values obtained are plotted on the line graph to decided the number of clusters.(Robert GOVe, 2017). Below figure represents an elbow graph.

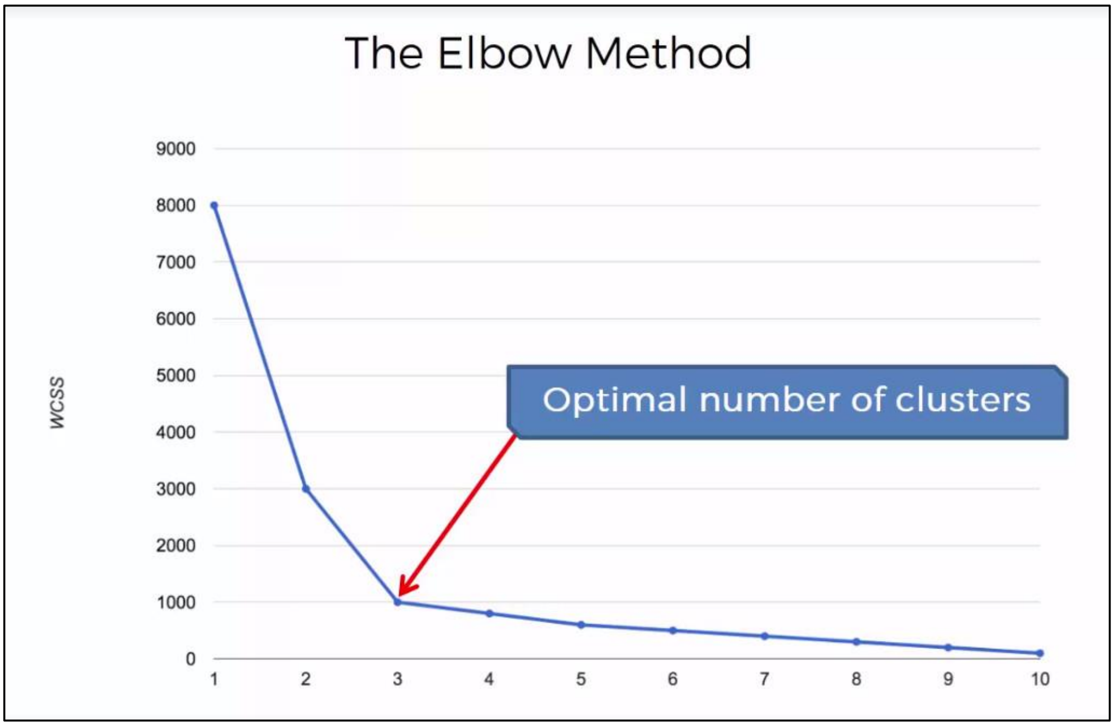


Figure 11 Elbow graph adopted from (Asank, 2017)

The reason why its name Elbow graph is because the line on it depicts a human arm and the point where there is an inclination denotes the elbow, and this inclination is the suggested number of clusters(Asank, 2017), there are cases when the dataset is not appropriate so in such cases there are possibilities of the elbow graph being not so precise in such case reconsidering the k value would be optimal to get the desired results(Robert GOVe, 2017)

2.9 Confusion Matrix and Classification Report:

A classification model's report is understood by the classification report(Shivam, 2019), this report shows the accuracy, precision, F-1score , Recall and support scores of the model for all the available classes, precision of the model here will let us know how much percent of the predicted outputs are correct, and the recall value gives us the knowledge of about how much predictions are accurate in each class, the F-1 score will let us know the amount of percent of accurate predictions irrespective of the classes, and the support will give us the idea of the occurences in each class from the dataset, generally there are four methods to find out the prediction accuracy which are False Positive, True Positive, False Negative and True Negative.(Shivam, 2019). Here's when confusion matrix comes into action, confusion matrix is a “it is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values”(Sarang, 2018); in this technique there’s a matrix where the size is MxM and the M is decided on the basis of output classes, this confusion matrix evaluates the performance of the classification model, if we take into account the 2x2 matrix which the classification model has 2 outputs then we will get 4 combinations such as FP,FN, TP and TN. if the TP section gets higher number of values than other sections then we can consider that the model has predicted the values more accurately, which means the higher the value of TP the more accurate is the prediction, and if the values fall under the FP then we get Type-1 Error whereas if the values are under FN then we will get Type-2 Errors(Sarang, 2018)

2.10 Previous work done under Recommendation systems:

the authors of this paper conducted a research on outfit recommendation system based on deep learning, (Huang and Huang, 2017) in this paper they classified the outfit in the sense whether it was good or bad, they divided the process into two parts where the first part comprised of feature extraction with the help of ResNet-50 and the other part consisted of the binary classification which classified the outfits using 2 layers of Multi-Layer Perceptron MLP, their system produced 1 or 0 when the annotated data along with image of cloth is passed as the input , where 1 denotes for good and 0 denotes for bad outfit.

(Guan, Qin and Long, 2019)this paper compared several different recommendation system and concluded the best model which gives the best accuracy compared to all, they used images , attributes which had information about the size, texture, pattern and design features and meanings of clothes which included casual, party, professional, trendy ,classic. Below figure shows the considered models:

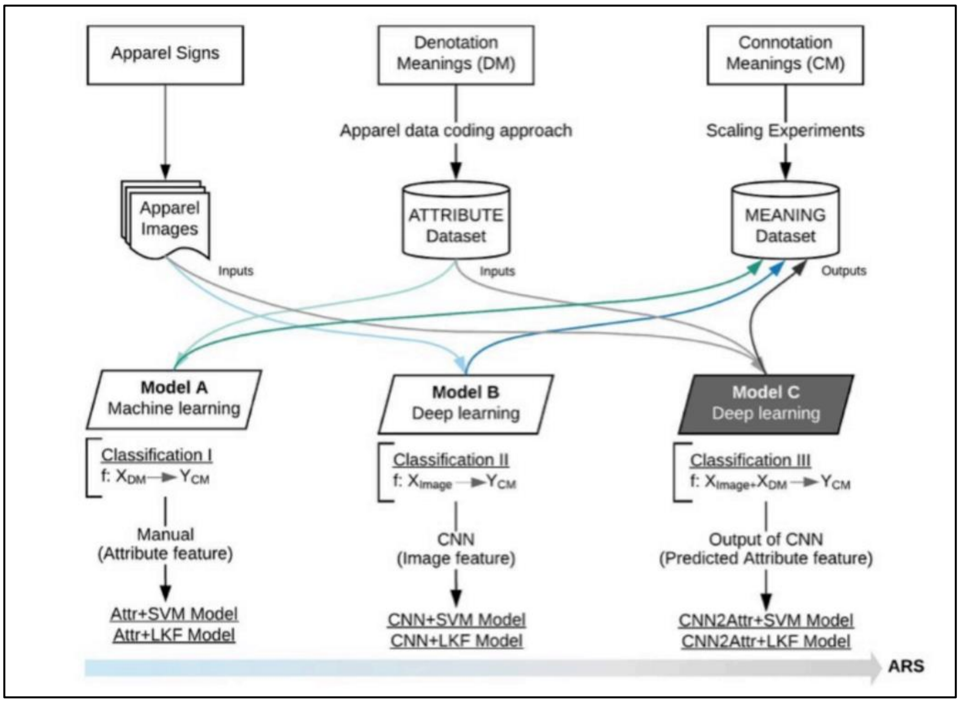


Figure 12 Different models architecture adopted from(Guan, Qin and Long, 2019)

As the figure depicts model A considers only attributes, and model B considers only images whereas model C takes both images and attributes.

Chapter 3 System Design

3.1 Chapter Overview:

In this chapter I am going to talk about the methods and approaches involved in this particular project, the overall research involves machine learning as well as deep learning algorithm which is implemented under supervised learning and unsupervised learning, initially the project starts with image processing techniques and after that splitting the data for training and testing purpose, second step is to implement algorithms and capture the results obtained, after the results obtained through implementing the algorithms I'll compare the results that is I'll compare the accuracy, performance, strengths and weaknesses, based on this comparison an algorithm will be identified that gives the better results in all the aspects which will be used further to predict the image class and recommend similar images. This project would be focusing on image processing, image identification, image classification in supervised learning whereas in unsupervised learning the focus will be on image clustering and feature learning, and clothes recommendation, in this section I'll also explain the different packages and technologies used during the implementation, this research will include the steps given below:

1. Data Acquisition and Data Balancing.
2. Deciding the technologies and packages to be used for Supervised learning and Unsupervised learning.
3. Image Processing, Data Extraction and Feature Engineering.
4. Build and Train the Supervised learning and Unsupervised learning algorithms.
5. Testing the Models on the test dataset and calculate the test accuracies.
6. Based on the prediction from the given input image recommending five apparels from the same Category.

3.2Data Acquisition:

As we all are aware to perform some analysis we require a set of data and these data are responsible for providing us with some useful insights, so for this project I have taken the dataset from kaggle.com, https://www.kaggle.com/trolukovich/apparel-images-dataset this dataset was published by Alexander Antonov with the name "Apparel images Dataset", in this dataset there 11384 apparel images comprising of 5 different categories of apparels which includes shirts, pants, dresses, shorts and shoes. These images are available in 6 different colours Black, Brown, Blue, Green, White and Red, being precise this dataset has 2570 images of Dresses, 1686 images of Shirts, 2789 images of Pants,3418 images of shoes and 922 images of shorts and all of these images are in the ".jpg" format. Below figure shows some sample images of the same.

3.3 Technologies used:

I have used python programming language to perform image processing, image data extraction, feature extraction and to Build Models in the Jupyter notebook. And to perform these techniques several python packages needed to be installed, and import which are Keras, Tensorflow, PIL, Pandas, Glob, OS, Altair, Numpy, OpenCV and Scikit-Learn, in supervised learning I am going to implement Decision Tree Classifier, XGBoost Classifier, and Gradient Boosting Classifier, and 5 layered Convolutional Neural Network, and in unsupervised learning I am implementing 5 Layered Convolutional Neural Network and VGG16 for feature engineering and K-means Algorithm for clustering the images.

3.4 Data Extraction and Image processing:

Image pre-processing step is very important for achieving better results we understood this from literature review, so here also I am going to perform some image pre-processing techniques, I am going to perform the image processing for supervised learning and for unsupervised learning as well, because for supervised algorithms , pixel data are set as the input and labels are set as the outputs, so we need to pass the pixel information to the algorithms by extracting it and in case of unsupervised algorithm we need to do feature engineering in order to extract features from the image for identifying the image class, later these image features will be used in clustering algorithms to predict the classes without any labels. Our main task is to pass the pixel information to the classification algorithms and label them into their categories such as dress, pant, shirt, short and shoes. Thus, for doing so we need to store the pixel and labels into a data frame. For the purpose of data extraction there are several steps which need to be implemented, first step is to load the paths of all the images which means to load the location of the dataset in whichever directory it is saved, after this step we will convert the RGB images into grayscale images in order to reduce the dimensions(reducing the dimensions refers to compressing the image) and after this step we will resize the images into 30x30 size(pixel size) because there's a severe chance of these images being of different sizes, after these images are resized we need to extract pixel information from these images and for this purpose we will use python library Python Image Library(PIL) commonly known as pillow , as the size of these images are 30x30 we will be getting 900 pixel values which means 900 columns plus the label columns, in this dataset there are 11385 images which means 11385 rows so we will end up getting 11385x902 size of the data frame, the snapshot if the data frame is given below:

Out[19]:

	Unnamed: 0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel892	pixel893	pixel894	pixel895	pixel896	pixel897	pi
0	0	11	12	12	14	20	25	11	55	21	...	87	86	84	83	82	83	83
1	0	220	222	224	226	226	230	233	234	236	...	238	238	238	237	237	237	23
2	0	149	152	130	88	70	79	228	204	189	...	0	0	0	0	0	0	9
3	0	65	66	66	67	67	68	68	68	68	...	79	78	77	76	75	74	74
4	0	255	255	255	255	255	255	255	255	255	...	255	255	255	255	255	255	25
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11380	0	255	255	255	255	255	255	229	123	106	...	194	222	255	255	255	255	25
11381	0	255	255	255	255	255	255	255	255	255	...	232	244	237	227	255	255	25
11382	0	255	255	255	255	255	255	255	255	255	...	255	255	255	255	255	255	25
11383	0	246	247	215	137	163	189	245	241	143	...	229	232	232	232	232	232	23
11384	0	255	255	255	255	252	248	250	252	254	...	255	255	255	255	255	255	25

11385 rows × 902 columns

Image processing in the case of unsupervised learning is going to be different in order to deal with the algorithms, first of all , all the images are loaded and after that these images are shaped to 224wide x 224 height x 3 dimensions so that size differences between images aren't an issue after this step all the images are converted to grayscale images to reduce the unnecessary dimensions or layers of the images and after this step on these grayscale images edge detection is performed I order to identify the shapes. The below figure shows the edge detection:

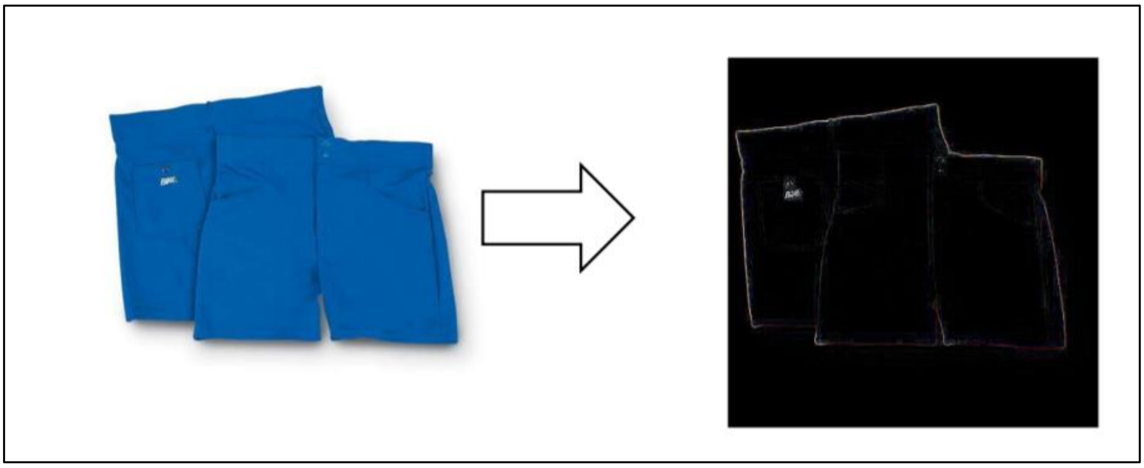


Figure 13 Expected output after edge detection

Edge detection is used to remove unnecessary information from the images and it takes important information into account such as shape of the object in an image, I have saved these images to an external folder as a backup in the next step , at last these images are stored into Numpy array and this array consists of pixel values where each pixel is of 3 bytes which represents the colour combination of RGB. After, this Numpy array is divided by 255 to bring all the pixel values ranging from 0-1 as a part of image normalization.

3.5 Training the SL Algorithms:

As we are going to compare the Supervised and Unsupervised learning algorithms, first we will apply machine learning and deep learning algorithms with the help of created .csv file mentioned in section 4.4, we already know that supervised learning is mostly used for regression and classification purpose from the study of literature review, the data available for the supervised learning should be labelled with categorical values being more precise what it mean is there should be a relationships between input and output variables during the training process, so when at testing phase and while validating the model, the predictions are made on the basis of input variables on the unseen data and will provide the required output.

For the purpose of image Classification I am implementing Decision tree classifier, XG Boost Classifier and Gradient Boosting classifier in Machine Learning whereas 5 Layered

Convolutional Neural Network in deep Learning, these algorithms are chosen because they help to improve accuracy, user friendly and easy to implement, at last the results of these algorithms would be compared to identify the best algorithm for classification.

3.6 Approach for Machine Learning Algorithms:

Observing and evaluating each model's accuracy and performance is our main objective.
For all the machine learning algorithms the generated .csv file is given as the input to train the models, this csv file has 11385 rows where each row represents an image from the dataset, this dataset is segregated into two parts for the purpose of training and testing using the functions of Scikit-Learn, moving forward traditionally, 80% of the data will be used for training purpose and rest 20% will be used for testing, the training dataset consists of input 901 columns, from these input columns the independent variables need to be passed to the data frame "X train" i.e. 900 columns, remaining one column is passed to the "Y_train", which is dependable variable i.e. the output data frame while training the models, the remaining 20%of the data which consists 900 input columns are passed to the "X_test" which will be used to predict the output values on the trained model, the actual output values of testing dataset will be passed to "Y_train"
Data frame, the accuracy of the model will be calculated by comparing the values of with the predicted values obtained through applying the "X_test " dataset on the model. Same method like this will be used for other Machine Learning algorithm as well.

3.7 Approach for Deep learning Algorithms:

Talking about the deep learning algorithm in this project 5 layered Convolutional Neural Network will be implemented and here there is no need of image pre-processing apart from resizing the images because the CNN model is implemented under supervised learning, if the data in the images are more than the training time taken will be more keeping this aspect in mind I will resize all the images in 64x64 size, one condition for the CNN is that all the images that need to be separated and it should be sorted according to its different categories and folders, so here I am using ImageDataGenertor, it is a fucntion which image pre-processing package of keras, used to create the dataset of training and testing with their labels so there is no need to create the labels manually, once this process of loading the images into training

and testing datasets, these images are sent to 5 Layered Convolutional Neural Network, this CNN does the image classification as well as identification of the images, the feature learning here would be like this, as this CNN has multiple layers in it which are Convolutional layers, Batch-Normalization layers, 2-D Max-pooling layers, Dropout layers, Flattening layer, Dense layers and Fully connected layer, this model is added with a convolutional layer with 32 filters and along with the activation fucntion of ReLu at the base followed accompanied by the two more convolutional layers with 64 filters each and again followed by two convolutional layers with 128 filters each, batch normalization, max pooling and dropout measures are executed in between these convolutional layers. After these steps are executed the features of the images are extracted and once these features are extracted they are passed to the algorithms to classify the images after the this step the classification takes place and in classification firstly all the features are flattened, after that the fully connected layer is applied for the purpose of connecting the image features to recognise the patterns. After this step the Softmax layer comes into action which is a dense layer which has sigmoid as the activation fucntion and it has 5 filters which classifies the images into their respective image classes. And at last all these layers are assembled and compiled by using the adam optimiser function, binary crossentropy as the loss fucntion and accuracy as metrics, using both training dataset and testing dataset the model is built, we will get both training and testing accuracy which is also called validation accuracy after each and every epoch is completed by this algorithm,
In the beginning ill be training this algorithm with 5 epochs and if the accuracies aren't satisfactory or not meeting the benchmark the epoch size will be doubled the prior epochs.

3.8 Training the Unsupervised learning Algorithms:

We understood that unsupervised learning is not dependent on the labelled data but instead it is dependent on the image patterns from the study of literature survey so the main objective of using unsupervised learning is to find out if there is any new category of images coming up which is not the part of this existing apparel category then this unsupervised learning model has the ability to learn the image features and cluster it together into new category. So for doing this I am using labels of the apparels that is apparel names to train the model, the overall approach involves three stages which are image processing, Feature Extraction and Image Clustering, as a first step we will move all the available images to single folder to make algorithm learn the objects shapes in the images. Here I am going to use 3 algorithms : K-means clustering algorithm, CNN with K-means and VGG16 with K-means algorithm. To perform image processing I am going to convert RGB image format to RGB to avoid differences in the images,

After that all the images are resized to 224x224 size using the OpenCV python package. After this step next step is for extracting the pixel values from all the images and storing it to a 3-dimensional Numpy array and at last all the images are divided by 255 for normalization purpose here we complete our image processing step. Later if the implemented algorithms doesn't gives the expected results we will go for edge detection in order to reduce the unnecessary information from the images.

3.9 Feature Extraction:

Feature extraction process comes after the image processing, in this process we will pass the NumPY array as an input to the deep learning algorithms. In this project I am considering 5 layered convolutional neural network along with VGG16 a pre-trained transfer learning algorithm to extract the features and store those features into one dimensional vectorization form and after this later these features are passed as an input for K-means clustering algorithm.

3.10 Image Clustering:

After the feature extraction process, to identify the apparel classes these vectors are passed to clustering algorithm, in this case K-means clustering algorithm is used to identify the apparel classes and to segment them to their apt groups, in this project comparison is going to be done between the K-means, 5 layered CNN with K-means and VGG16 with K-means algorithm for the purpose of identifying the best model for clustering. In order to calculate the accuracies Silhoutte score method is used and for performing clustering under machine learning k-means algorithm is used, firstly the pixels from the images are extracted and are normalised, by dividing each pixel by 255, in the next step these pixels dimensions are reduced to 2 components with the help of PCA technique, for implementing the K-means algorithm it is mandatory to decide upon the K value because the K value stands for clusters so if we see in our dataset there are 5 apparel classes therefore the K value must be 5, to be more sure on the clusters a statistical method which is a elbow method is going to be implemented in order to just be sure about the clusters, after getting the K value K-means algorithm would be implemented using K number of clusters, in deep learning I am using 5 layered convolutional neural network algorithm along with the most popular VGG16 for extracting the features from the images and later on the extracted feature K-means algorithm is used for image clustering.

3.11 Image Recommendation:

After the comparison of all the supervised and unsupervised algorithms I am going to finalize one algorithm those which gives better accuracy, to procced further with recommendation we need to consider few steps here such as : A dataframe is implemented which will act as the database for research purpose after that to recommend the images of same category from the database a python fucntion is coded, below figure shows an expected result of the recommendation.

Input image *Expected output images*

Chapter 4 Results and Findings

4.1 Chapter Overview:

In this chapter I am going to talk about the results I acquired from the Machine Learning and Deep learning algorithms carried out under Supervised and Unsupervised Learning, precisely I'll be discussing about the results gained from Supervised Learning(Decision Tree Classifier, Gradient Boosting Algorithm, XGBoost classifier, and 5 layered CNN) and in case of Unsupervised Learning(K-Means, CNN integrated with K-means and VGG16 integrated with K-means).

4.2 Supervised Learning Algorithms:

Under Supervised Learning three Machine Learning algorithms and one Deep Learning algorithm is implemented, in case of Machine Learning Algorithms metrics such as model's accuracy, recall, precision value, f1 score and confusion matrix are taken into account whereas in case of Deep Learning Validation loss, Training loss, Validation accuracy, and model's accuracy are considered.

For Decision tree Classifier the accuracy score is around 69% and weighted average of recall, precision and F-1 score is around 70%. Comparing the score of five available categories, the category of "pant" tops the list in precision, recall and F-1 score which is 75%,76% and 75%, while the category "shorts" scores the lowest with 39% precision score, 45% recall score and 42% F-1 score. While the scores of remaining categories ranges from 65-76%,The confusion matrix shows predicted accuracy of the images, it shows that 334 images of dress, 418 of pants, 254 of shirts,511 of shoes and 76 of shorts were predicted accurately.

The below image is an output of Decision Tree classifier:

```
Accuracy:  0.6996047430830039
------------------------------------------------------------
Classification Report
---------------------------------------
              precision    recall  f1-score   support

       dress       0.67      0.65      0.66       511
       pants       0.75      0.76      0.75       553
       shirt       0.74      0.75      0.74       338
       shoes       0.76      0.72      0.74       706
      shorts       0.39      0.45      0.42       169

    accuracy                           0.70      2277
   macro avg       0.66      0.67      0.66      2277
weighted avg       0.70      0.70      0.70      2277

------------------------------------------------------------
Confusion Matrix
---------------------------------------
[[334  54  35  67  21]
 [ 52 418   8  44  31]
 [ 33  18 254  15  18]
 [ 67  49  29 511  50]
 [ 14  22  18  39  76]]
```

Figure 14 Decision tree algorithm results.

In case of Gradient Boosting classifier the overall accuracy obtained is 83%, weighted average of precision, recall and F-1 score is 83%, here also the category “pants” tops the list by acquiring the score of 88%,84%,86% in precision, recall and F-1 score, the lowest score is scored by shorts which is 80%,61,% and 69% in terms of precision, recall and F-1, other categories ranges between 85-89% the confusion matrix shows this model predicted 436 images of dress, 467 of pants, 290 of shoes, 600 of shorts and 103 of shoes accurately.

Below is the overall result of Gradient Boosting Classifier:

```
Accuracy:  0.8326745718050066
----------------------------------------------------------
Classification Report
---------------------------------------
              precision    recall  f1-score   support

       dress       0.78      0.85      0.81       511
       pants       0.88      0.84      0.86       553
       shirt       0.87      0.86      0.86       338
       shoes       0.83      0.85      0.84       706
      shorts       0.80      0.61      0.69       169

    accuracy                           0.83      2277
   macro avg       0.83      0.80      0.81      2277
weighted avg       0.83      0.83      0.83      2277

----------------------------------------------------------
Confusion Matrix
---------------------------------------
[[436  13  18  42   2]
 [ 37 467  11  32   6]
 [ 25   3 290  15   5]
 [ 52  33   9 600  12]
 [ 11  17   7  31 103]]
```

Figure 15 Result of gradient boosting algorithm

XG boost classifier showed accuracy of 87% wherein the category shirt topped the list here in terms of precision value, recall and F-1 score which is 91%,88% and 90% respectively, as there is seen the category shorts is lagging behind the most compared to other categories with the precision score of 86% and recall score of 70% and F-1 score of 78% , other categories lies in the range of 80-90%. The confusion matrix here shows that 457 of dress, 490 of pants,299 of shirt ,632 of shoes and 119 of shorts images are accurately predicted.

Below is the snapshot of overall result of XG Boot algorithm:

```
Accuracy:  0.8770311813790075
-----------------------------------------------------------------
Classification Report
---------------------------------------
              precision    recall  f1-score   support

       dress       0.85      0.89      0.87       511
       pants       0.89      0.89      0.89       553
       shirt       0.91      0.88      0.90       338
       shoes       0.87      0.90      0.88       706
      shorts       0.86      0.70      0.78       169

    accuracy                           0.88      2277
   macro avg       0.88      0.85      0.86      2277
weighted avg       0.88      0.88      0.88      2277

-----------------------------------------------------------------
Confusion Matrix
---------------------------------------
[[457  16  13  23   2]
 [ 19 490   8  31   5]
 [ 18   3 299  13   5]
 [ 39  23   5 632   7]
 [  7  17   2  24 119]]
```

Figure 16 Overall Result of XG Boost Algorithm

Under the Deep learning algorithm I implemented 5 layered Convolutional Neural Network and the accuracy is based on 4 metrics: Train accuracy, Train Loss, Validation accuracy (Test accuracy) and validation loss. The first training of CNN is done with 5 epochs, with this the validation loss is reduced from infinity to 0.04721 which is approximately 4% loss at the end of training, the training accuracy is 95% and the validation accuracy is 97%, below figure gives us the idea about the accuracy improvement and the validation loss improvement after each epoch.

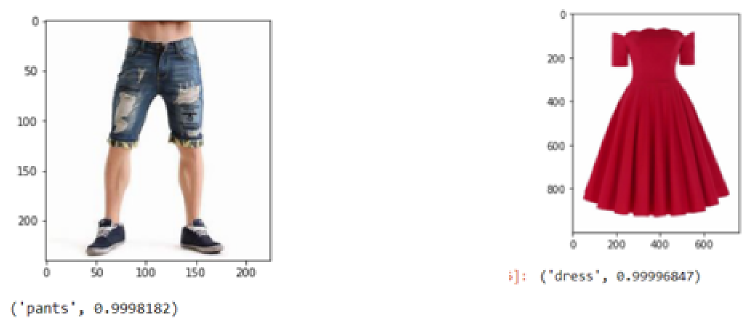
```
Epoch 1/5
346/346 [==============================] - 167s 484ms/step - loss: 0.3278 - accuracy: 0.8646 - val_loss: 0.2374 - val_accuracy:
0.9257

Epoch 00001: val_loss improved from inf to 0.23744, saving model to CNN_checkpoint
Epoch 2/5
346/346 [==============================] - 122s 354ms/step - loss: 0.1736 - accuracy: 0.9361 - val_loss: 0.2041 - val_accuracy:
0.9668

Epoch 00002: val_loss improved from 0.23744 to 0.20409, saving model to CNN_checkpoint
Epoch 3/5
346/346 [==============================] - 121s 349ms/step - loss: 0.1396 - accuracy: 0.9487 - val_loss: 0.1349 - val_accuracy:
0.9687

Epoch 00003: val_loss improved from 0.20409 to 0.13489, saving model to CNN_checkpoint
Epoch 4/5
346/346 [==============================] - 120s 348ms/step - loss: 0.1224 - accuracy: 0.9541 - val_loss: 0.0472 - val_accuracy:
0.9775

Epoch 00004: val_loss improved from 0.13489 to 0.04721, saving model to CNN_checkpoint
Epoch 5/5
346/346 [==============================] - 119s 345ms/step - loss: 0.1116 - accuracy: 0.9590 - val_loss: 0.0740 - val_accuracy:
0.9779

Epoch 00005: val_loss did not improve from 0.04721
```

Figure 17 CNN trained with 5 epoch

After training CNN with 5 epochs, later the epochs are increased to 10 from 5 in order to reduce the loss and increase the train and test accuracies, training accuracy is 97% at the end of 10 epochs and validation accuracy is 97% and the validation loss is reduced to 0.2% after 10 epochs below figure shows the result of CNN algorithm trained with 10 epochs.

```
Epoch 00005: val_loss improved from 0.05051 to 0.01247, saving model to CNN_checkpoint
Epoch 6/10
346/346 [==============================] - 116s 336ms/step - loss: 0.1033 - accuracy: 0.9617 - val_loss: 0.0550 - val_accuracy:
0.9774

Epoch 00006: val_loss did not improve from 0.01247
Epoch 7/10
346/346 [==============================] - 116s 336ms/step - loss: 0.0955 - accuracy: 0.9647 - val_loss: 0.0557 - val_accuracy:
0.9811

Epoch 00007: val_loss did not improve from 0.01247
Epoch 8/10
346/346 [==============================] - 117s 338ms/step - loss: 0.0880 - accuracy: 0.9673 - val_loss: 0.0768 - val_accuracy:
0.9817

Epoch 00008: val_loss did not improve from 0.01247
Epoch 9/10
346/346 [==============================] - 117s 337ms/step - loss: 0.0794 - accuracy: 0.9711 - val_loss: 0.0399 - val_accuracy:
0.9834

Epoch 00009: val_loss did not improve from 0.01247
Epoch 10/10
346/346 [==============================] - 117s 339ms/step - loss: 0.0817 - accuracy: 0.9710 - val_loss: 0.0012 - val_accuracy:
0.9777

Epoch 00010: val_loss improved from 0.01247 to 0.00125, saving model to CNN_checkpoint
```

After this CNN model is trained it is further used to predict the images which I downloaded from google platform and not from the test and train datasets, below is the snapshot of the predicted images and their accuracies.



CNN predictions on unseen images.

4.2.1 Insights from Supervised Learning Algorithms:

From the results obtained from the training of algorithms under Machine Learning and Deep Learning, the XG Boost classifier implemented along with GridSearchCV tops the list with 89% accuracy whereas in Deep Learning we saw that 5 layered Convolutional Neural Network trained with 10 epochs has the accuracy of 97% with validation accuracy around 98%, so in Supervised Learning algorithms under Machine Learning XGBoost classifier gives better accuracy compared to other algorithms, and which is followed by the 5 layered CNN in Deep Learning, whereas we saw that Decision Tree Classifier under Machine learning did not give better accuracy so we can conclude that Decision Tree Classifier cannot be used in the case of such apparel prediction problem.

Algorithms	Overall Accuracy %
Decision Tree Classifier	67
Gradient Boosting Algorithm	83
XGBoost Classifier	87
XGBoost Classifier with GridSearchCV	87
Convolutional Neural Network with 5 epochs	95
Convolutional Neural Network with 10 epochs	97

4.3 Unsupervised Learning Algorithms:

In Unsupervised Learning I have implemented three algorithms under Machine Learning and Depp learning algorithms, which is K-means clustering algorithm, CNN along with K-means and VGG16 integrated with K-means algorithm, each of the models accuracy is calculated with help of Silhouette score, each cluster is identified with the help of this score and the average value of these clusters score is considered as the overall accuracy of the model.

Starting with the K-means algorithm before proceeding with this algorithm there's a need of deciding the K value, we have 5 categories here so apparently the value should be 5 but just for being more precise in choosing the value Elbow method is been implemented to validate potential number of clusters, below figure shows the elbow graph which shows a bend at k value 5 , which means that 5 clusters could eb formed through the dataset we have.

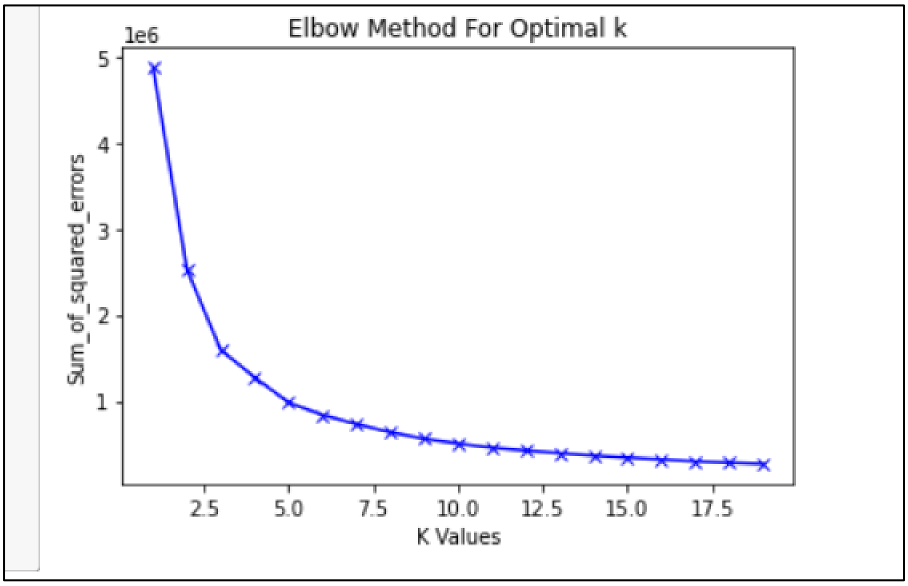
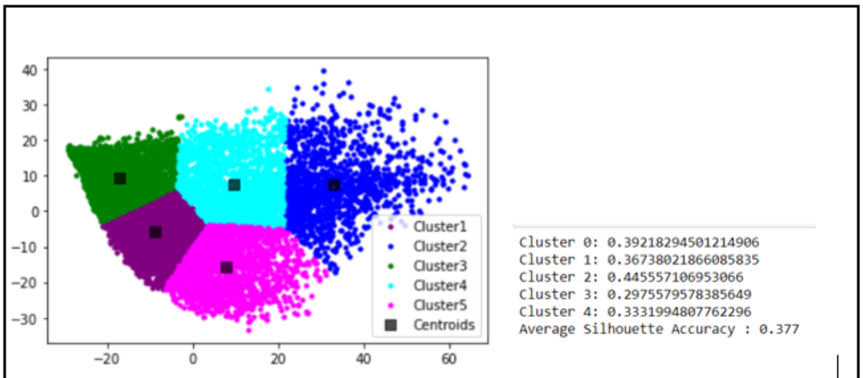


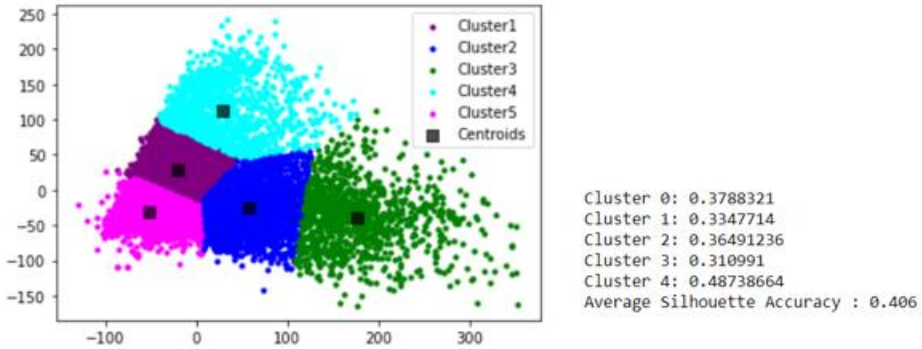
Figure 18 Identifying optimal value for i.e. the K value

K-means Machine Learning algorithm, the model is implemented with the raw pixel values extracted from the 11834 images, the overall accuracy of the ,model that is Silhouette score is 37% and the silhouette sample scores for clusters 0,1,2,3,4 are 39%,36%,44%,29%and 33% approx. respectively, from the below figure we can see that there is each datapoint for particular clusters which represents each image class, also it shows how different image class is clustered differently.



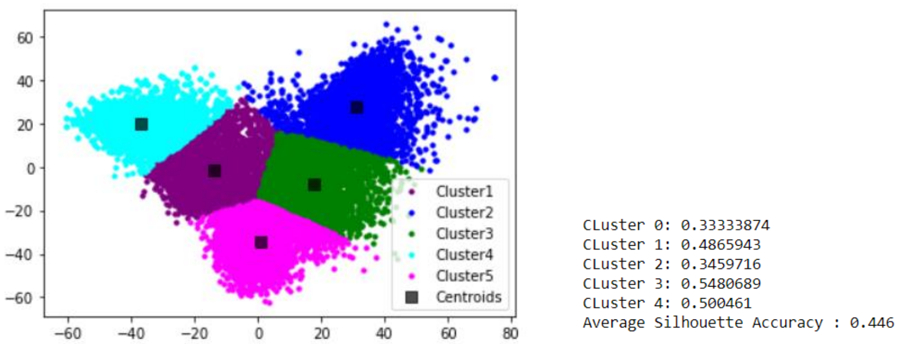
Clustering graph of K-means along with the obtained score

As the K-means algorithm's accuracy score is below the line that is 40% further for extraction of features 5 layered CNN with VGG16 is used and implemented, whereas in second model CNN is integrated with K-means algorithm, where CNN will be used for feature learning purpose and K-means is used to cluster the images, the average accuracy that is Silhouette score is 40% and the Silhouette scores of the clusters 0,1,2,3,4 are 37%,33%,36%,31% and 48% approx. as shown in the below figure.



Clustering graph of CNN with K-means along with the Silhouette score

VGG16 integrated with K-means algorithm is our third model, where VGG16 will extract the features and K-means will cluster the images the overall accuracy (Silhouette score) Obtained is 44% and Silhouette sample scores for the clusters 0,1,2,3,4 are 33%,48%34% and 54% respectively, below figure shows the same.



Clustering graph of VGG16 integrated with K-means

4.3.1 Insights from Unsupervised Learning Algorithms:

Three models were implemented under the Unsupervised learning algorithms, after observing the results of each model, the accuracies acquired by the models are 37%,40% respectively, which is not even close to 50%, the reason to get such accuracy is because of the size of the dataset, this dataset has 11384 images which is approximately 2276 image per category, it is very difficult for an algorithm to understand the features from so small amount of data and without any labels, which leads to poor results so for getting better results we need to improve the dataset and get more amount of data which means that we need to increase number of images per category under Unsupervised Learning.

Chapter 5 EVALUATION

In this chapter the research question which was proposed at the beginning is answered and a detailed explanation is given.

5.1Research Questions:

1. Can we predict the apparel class and recommend the similar apparels using images alone as an input and to what extent?

From the results obtained through data analysis it is observed that, it is possible to recommend apparels on the basis of images. In chapter 5 we discuss the performance, overall results and accuracies obtained by different algorithms,

Among all those algorithms the 5 Layered CNN acquired the better and higher accuracy compared to other algorithms so we can conclude that this model is best suited to recommend apparels. The best thing about this model is that it doesn't need any textual data to process but only needs images as input to recommend the apparels, because this model is implemented under supervised learning algorithm, labels are passed as input to the algorithm, the CNN model will automatically pick up the apparel class which is apparently labels(apparel class) and no manual interference is required in doing so. Thus, the answer to the research question is yes, we can predict the images from the input image and recommend the apparels from the same class to which the input image belongs to.

2. Which Algorithm suits best for recommending the apparels using the image dataset under Machine learning and Deep Learning?

Through observation of results of all the algorithms we can conclude that the supervised learning algorithm approach suits the best for the type of dataset we used in this research, 5 Layered CNN under Deep learning suits the best for classification purpose, and under Machine Learning algorithm the XGBoost algorithm achieved 87% accuracy which is the best amongst all other machine learning algorithms. In the case of unsupervised learning approach VGG16 integrated with K-means clustering algorithm can be optimal if and only if the required or appropriate number of images are passed as an input to the model.

Chapter 6 Conclusions

In this research we implemented different models under Machine Learning and Deep Learning. We studied two approaches:-Supervised Learning and Unsupervised Learning. From the overall results we can conclude that we can form a viable base for the Fashion Apparel recommendation System. From the research we get the insights of which algorithms are best suitable for the given dataset, we identified that 5-Layered Convolutional Neural Network predicts with 97% accurately and we also found that Machine Learning algorithm can also predict better if given an image as input. The XGBoost algorithm proved that by acquiring 87% of accuracy in predicting the apparels of different class but in the case of Unsupervised learning, algorithms was not able to obtain higher accuracy because of lack of data in the dataset. The highest accuracy was obtained by VGG16 which is around 42% when implemented as feature extractor and integrated with K-means for clustering. Therefore we can conclude that to use Supervised Learning approach we need higher amount data to train the models.

6.1 Recommendations and Future work:

We have identified which model suits best for the recommendation. Looking to the future the model developed in this research could be used in real-time scenarios which will benefit a lot of companies in the fashion domain to identify the latest fashion trends and also it will help users to identify the products based on their likes. Talking about the advancement to the existing model, Real-time images could be captured through a camera and the images could be pre-processed and send it to the model to predict the image class and can thus show the users similar kind of apparels. This could benefit the organization in terms of improved sales and thus improved profit and customer satisfaction.

Appendices

Appendix A: References

'2019 - Content based Apparel Recommendation System for Fa.pdf' (no date).

Asank, P. (2017) 'Finding the optimal number of clusters for K-Means through Elbow method using a mathematical approach compared to graphical approach'.

'Canny - 1986 - A Computational Approach to Edge Detection.pdf' (no date).

carritech (2017) 'Carritech telecommunications'.

'Chen et al. - 2018 - Edge detection based on machine vision applying to.pdf' (no date).

'Content based Apparel Recommendation System for Fashion Industry' (2019) International Journal of Engineering and Advanced Technology, 8(6), pp. 509–516. doi: 10.35940/ijeat.F7880.088619.

'deep-learning-using-transfer-learning-python-code-for-resnet50-8acdfb3a2d38' (no date). Available at: https://towardsdatascience.com/deep-learning-using-transfer-learning-python-code-for-resnet50-8acdfb3a2d38 (Accessed: 24 August 2020).

'difference between ai and ml' (no date). Available at: https://medium.com/towards-artificial-intelligence/differences-between-ai-and-machine-learning-and-why-it-matters-1255b182fc6 (Accessed: 16 August 2020).

Eapen (2014) 'MEDICAL IMAGE SEGMENTATION FOR ANATOMICAL KNOWLEDGE EXTRACTION', Journal of Computer Science, 10(7), pp. 1253–1258. doi: 10.3844/jcssp.2014.1253.1258.

GeeksforGeeks (no date) 'RGB image representation'.

Guan, C. et al. (2016) 'Apparel recommendation system evolution: an empirical review', International Journal of Clothing Science and Technology, 28(6), pp. 854–879. doi: 10.1108/IJCST-09-2015-0100.

Guan, C., Qin, S. and Long, Y. (2019) 'Apparel-based deep learning system design for apparel style recommendation', International Journal of Clothing Science and Technology, 31(3), pp. 376–389. doi: 10.1108/IJCST-02-2018-0019.

'Hesamian et al. - 2019 - Deep Learning Techniques for Medical Image Segment.pdf' (no date).

Hu, Y., Yi, X. and Davis, L. S. (2015) 'Collaborative Fashion Recommendation: A Functional Tensor Factorization Approach', in Proceedings of the 23rd ACM international conference on Multimedia - MM '15. the 23rd ACM international conference, Brisbane, Australia: ACM Press, pp. 129–138. doi: 10.1145/2733373.2806239.

Huang, Y. and Huang, T. (2017) 'Outfit Recommendation System Based on Deep Learning', in Proceedings of the 2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017). 2nd International Conference on Computer Engineering, Information Science & Application

Technology (ICCIA 2017), Wuhan, China: Atlantis Press. doi: 10.2991/iccia-17.2017.26.

'image proccesing.pdf' (no date).

'image-types.html' (no date). Available at: https://in.mathworks.com/help/matlab/creating_plots/image-types.html (Accessed: 25 August 2020).

'INSTANCE SEGMENTATION GENERATOR FOR FASHION IMAGES.pdf' (no date).

j brownlee (2016) 'supervised vs unsupervised'.

'Jeyalaksshmi and Prasanna - 2017 - Measuring distinct regions of grayscale image usin.pdf' (no date).

Kalra, B., Srivastava, K. and Prateek, M. (2016) 'Computer vision based personalized clothing assistance system: A proposed model', in 2016 2nd International Conference on Next Generation Computing Technologies (NGCT). 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India: IEEE, pp. 341–346. doi: 10.1109/NGCT.2016.7877438.

Kang, W.-C. et al. (2017) 'Visually-Aware Fashion Recommendation and Design with Generative Image Models', in 2017 IEEE International Conference on Data Mining (ICDM). 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA: IEEE, pp. 207–216. doi: 10.1109/ICDM.2017.30.

'Károly et al. - 2018 - Unsupervised Clustering for Deep Learning A tutor.pdf' (no date).

KD nuggets (no date) 'What is support vector machine and why would i use it?'

Kim, K. G. (2016) 'Book Review: Deep Learning', Healthcare Informatics Research, 22(4), p. 351. doi: 10.4258/hir.2016.22.4.351.

'Kumar and Sharma - 2016 - Approaches, Issues and Challenges in Recommender S.pdf' (no date).

Kumar, B. and Sharma, N. (2016) 'Approaches, Issues and Challenges in Recommender Systems: A Systematic Review', Indian Journal of Science and Technology, 9(47). doi: 10.17485/ijst/2015/v8i1/94892.

'machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761' (no date). Available at: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 (Accessed: 26 August 2020).

mark ryan (2015) 'supervised learning'.

'McREYNOLDS and Blythe - Illustration and Artistic Techniques.pdf' (no date).

van Meteren, R. and van Someren, M. (no date) 'Using Content-Based Filtering for Recommendation', p. 10.

'Nethaji and Shanmugasundaram - 2020 - The analysis and manipulation of a digitized image.pdf' (no date).

'Raju and Karthikeyani - 2012 - Improved Satellite Image Preprocessing and Segment.pdf' (no date).

'Rawat and Wang - 2017 - Deep Convolutional Neural Networks for Image Class.pdf' (no date).

Rawat, W. and Wang, Z. (2017) 'Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review', Neural Computation, 29(9), pp. 2352–2449. doi: 10.1162/neco_a_00990.

Robert GOVe (2017) 'Using the elbow method to determine the optimal number of clusters for k-means clustering'.

Sarang, N. (2018) 'Understanding Confusion Matrix'.

Shivam, kohli (2019) 'Understanding a Classification Report For Your Machine Learning Model'.

shopifyplus (2020) 'ecommerce fashion industry staistics'.

'Simard et al. - 2003 - Best practices for convolutional neural networks a.pdf' (no date).

'the-architecture-and-implementation-of-vgg-16-b050e5a5920b' (no date). Available at: https://medium.com/towards-artificial-intelligence/the-architecture-and-implementation-of-vgg-16-b050e5a5920b (Accessed: 23 August 2020).

'understand-implement-resnets' (no date). Available at: https://cv-tricks.com/keras/understand-implement-resnets/ (Accessed: 24 August 2020).

'understanding CNN' (no date). Available at: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148 (Accessed: 23 August 2020).

'unsupervised machine learning webpage' (2019).

'what-is-the-difference-between-computer-vision-and-image-processing' (no date). Available at: https://analyticsindiamag.com/what-is-the-difference-between-computer-vision-and-image-processing/#:~:text=Image%20processing%20is%20a%20subset%20of%20computer%20vision.,to%20recognise%20objects%2C%20defect%20for%20automatic%20driving%2C%20 (Accessed: 17 August 2020).

Xu, R. and WunschII, D. (2005) 'Survey of Clustering Algorithms', IEEE Transactions on Neural Networks, 16(3), pp. 645–678. doi: 10.1109/TNN.2005.845141.

Zhang, S. et al. (no date) 'Deep Learning based Recommender System: A Survey and New Perspectives', ACM Computing Surveys, 1(1), p. 35.

APPENDIX B: IMPLEMENTED CODE FOR THIS RESEARCH

```
import PIL
from PIL import Image
import pandas as pd
import glob
import os
import altair as alt
import seaborn as sns
import numpy as np
import re
```

```
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
```

```
import numpy as np
import keras
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
from keras.models import Sequential
from keras.layers import Conv2D, Dense, Flatten, MaxPooling2D
from keras.layers import BatchNormalization, Activation, Dropout
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint
```

```
import sys
from PIL import Image
sys.modules['Image'] = Image
import cv2 as cv
import tensorflow as tf
import keras.backend as K
```

```
from keras import regularizers
from time import time
from sklearn.cluster import KMeans
from keras import callbacks
from keras.models import Model
from keras.optimizers import SGD
from keras.layers import Dense, Input
from keras.initializers import VarianceScaling
from keras.engine.topology import Layer, InputSpec
import sklearn.metrics as metrics
from sklearn.metrics import accuracy_score, normalized_mutual_info_score
from keras.layers import Dense,GlobalAveragePooling2D
from keras.preprocessing.image import load_img
from keras.applications.vgg16 import VGG16
```

```
image_path=glob.glob("apparel dataset//*//*")
```

```
category_names=[]
for f in glob.glob("apparel dataset//*//*"):
    path = os.path.split(f)[-2]
    category_names.append(path)
```

```
categories=[]
for i in range(0,len(category_names)):
    categories.append(category_names[i].split("\\""")[1].split('_')[1])
```

```
database_dict =  {'Category':categories,'path':image_path}
database = pd.DataFrame(database_dict)
```

The below code will iterate through the dataset to extract the pixels. Below are the steps involved:

1. RGB image is converted to Grayscale
2. Grayscale image is resized to 30x30.
3. Image Pixels are extracted and appended to "data" dataframe. Each row consists pixels of one Image.
4. Column names are created, here the column names will start from pixel1 to pixel 900 and these values are stored in "Columns_names" list.
5. Category column which is created from "categories" list is appended to the "data" dataframe.

```
i=0
for img in image_path:
    raw_image = Image.open(img).convert("LA")
    grey_scale_image=raw_image.resize((30,30))
    pixels = list(grey_scale_image.getdata())
    if i==0:
        column_names =[]
        counter = 0
        for pixel in pixels:
            counter += 1
            column_names.append('pixel'+ str(counter))
        pixel_data =[]
        counter = 0
        for pixel in pixels:
            pixel_data.append(pixels[counter][0])
            counter += 1
        pixel_data = [pixel_data]
        data=pd.DataFrame(pixel_data)
        i = 1
    elif i==1:
        pixel_data =[]
        counter = 0
        for pixel in pixels:
            pixel_data.append(pixels[counter][0])
            counter += 1
          # pixel_data
        pixel_data = [pixel_data]
        data=data.append(pixel_data,ignore_index=False)
```

```
data["Category"]= categories
column_names.append("Category")
data.columns = column_names
```

```
data.to_csv("pixels.csv")
```

```
data_csv = pd.read_csv("pixels.csv")
```

```
data1 = data_csv.copy()
```

```
data1
```

```
#data1 = data1.drop("Unnamed: 0",axis=1)
```

```
X = data1.loc[:, ~data1.columns.isin(['Category'])]
Y = data1.loc[:, data1.columns == 'Category']

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.20, random_state=101)
```

```
# Shape of the 4 created datasets
print(X_train.shape)
print(Y_train.shape)
print(X_test.shape)
print(Y_test.shape)
```

```
from sklearn import preprocessing
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import accuracy_score, log_loss,classification_report,confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier
from sklearn.model_selection import GridSearchCV
from xgboost import plot_tree
from sklearn.preprocessing import OneHotEncoder
import random, shutil
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

```
#1. Decision tree classifier
decision = DecisionTreeClassifier()
decision.fit(X_train, Y_train)
DecisionTreeClassifier_predictions = decision.predict(X_test)

acc = accuracy_score(Y_test, DecisionTreeClassifier_predictions)
print("Accuracy:  "+str(acc))
print("---------------------------------------------------------------------------------")
print("Classification Report")
print("---------------------------------------")
print(classification_report(Y_test,DecisionTreeClassifier_predictions))
print("---------------------------------------------------------------------------------")
print("Confusion Matrix")
print("---------------------------------------")
print(confusion_matrix(Y_test,DecisionTreeClassifier_predictions))
```

```
# 2. Creating GradientBoostingClassifier
Gradient = GradientBoostingClassifier()
Gradient.fit(X_train, Y_train)
GradientBoostingClassifier_predictions = Gradient.predict(X_test)

acc = accuracy_score(Y_test, GradientBoostingClassifier_predictions)
print("Accuracy:  "+str(acc))
print("---------------------------------------------------------------------------------")
print("Classification Report")
print("---------------------------------------")
print(classification_report(Y_test,GradientBoostingClassifier_predictions))
print("---------------------------------------------------------------------------------")
print("Confusion Matrix")
print("---------------------------------------")
print(confusion_matrix(Y_test,GradientBoostingClassifier_predictions))
```

```
# 3. XGBoost Classifier
XGBmodel = XGBClassifier()
XGBmodel.fit(X_train, Y_train)
XGBmodel_prediction = XGBmodel.predict(X_test)


acc = accuracy_score(Y_test, XGBmodel_prediction)
print("Accuracy:  "+str(acc))
print("---------------------------------------------------------------------------------")
print("Classification Report")
print("---------------------------------------")
print(classification_report(Y_test,XGBmodel_prediction))
print("---------------------------------------------------------------------------------")
print("Confusion Matrix")
print("---------------------------------------")
print(confusion_matrix(Y_test,XGBmodel_prediction))
```

```
from sklearn.model_selection import GridSearchCV
XGBmodel = XGBClassifier()

optimization_dict = {'max_depth': [2,4,6,8],
                     'n_estimators': [50,100,200,400]}

XGBmodel = GridSearchCV(XGBmodel, optimization_dict,
                     scoring='accuracy', verbose=1)
XGBmodel.fit(X_train, Y_train)
```

```
XGBmodel_prediction = XGBmodel.predict(X_test)

acc = accuracy_score(Y_test, XGBmodel_prediction)
print("Accuracy:  "+str(acc))
print("---------------------------------------------------------------------------------")
print("Classification Report")
print("---------------------------------------")
print(classification_report(Y_test,XGBmodel_prediction))
print("---------------------------------------------------------------------------------")
print("Confusion Matrix")
print("---------------------------------------")
print(confusion_matrix(Y_test,XGBmodel_prediction))
```

```python
# dress
raw_image = Image.open("CoralaDress.jpg").convert("LA")
im=Image.open("CoralaDress.jpg")
grey_scale_image=raw_image.resize((30,30))
pixels = list(grey_scale_image.getdata())
pixel_data =[]
counter = 0
for pixel in pixels:
    pixel_data.append(pixels[counter][0])
    counter += 1
    # pixel_data
pixel_data = [pixel_data]
data2=pd.DataFrame(pixel_data)
data2.columns=column_names[0:900]
test_output = XGBmodel.predict(data2)[0][0:6]
plt.imshow(im)
plt.show()
print("Output: "+test_output)
```

```python
print("**********************Input*********************")
path = "manshort1.JFIF"
input_image = Image.open(path)
plt.imshow(input_image)
plt.show()
raw_image = Image.open(path).convert("LA")
grey_scale_image=raw_image.resize((30,30))
pixels = list(grey_scale_image.getdata())
pixel_data =[]
counter = 0
for pixel in pixels:
    pixel_data.append(pixels[counter][0])
    counter += 1
    # pixel_data
pixel_data = [pixel_data]
data2=pd.DataFrame(pixel_data)
data2.columns=column_names[0:900]
test_output = XGBmodel.predict(data2)[0][0:6]
print("**Output**")
recommend=database.query("Category == ""'"+test_output+"'""").sample(5).reset_index()
#recommed =
inc=0
for path in recommend["path"]:
    im = Image.open(recommend["path"][inc])
    plt.imshow(im)
    plt.show()
    inc +=1
```

```python
# All the images are loaded into the "training_set" variable using ImageDataGenerator functions from Keras packages
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)
training_set = train_datagen.flow_from_directory('CNN//train',
                                                 target_size = (64, 64),
                                                 batch_size = 32,
                                                 class_mode = 'categorical')
```

```python
# Initializing the size of the input images
input_shape = (64, 64, 3)
```

```
# Creating the 5 Convolutional Layered model architecture
CNN_model = Sequential()

# First Layer
CNN_model.add(Conv2D(filters=32, kernel_size=3, padding = 'same', input_shape = [64, 64, 3], kernel_initializer = 'he_normal', ac
CNN_model.add(BatchNormalization())
CNN_model.add(MaxPooling2D(3))
CNN_model.add(Dropout(0.25))

# Second Layer
CNN_model.add(Conv2D(filters=64,kernel_size= 3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'))
CNN_model.add(BatchNormalization())

# Third Layer
CNN_model.add(Conv2D(filters=64,kernel_size= 3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'))
CNN_model.add(BatchNormalization())
CNN_model.add(MaxPooling2D())
CNN_model.add(Dropout(0.25))

# Fourth Layer
CNN_model.add(Conv2D(filters=128, kernel_size=3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'))
CNN_model.add(BatchNormalization())

# Fifth Layer
CNN_model.add(Conv2D(filters=128,kernel_size= 2, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'))
CNN_model.add(BatchNormalization())
CNN_model.add(MaxPooling2D())
CNN_model.add(Dropout(0.25))

# Fully Connected Layer
CNN_model.add(Flatten())
CNN_model.add(Dense(1024, activation = 'relu', kernel_initializer = 'he_normal'))
CNN_model.add(BatchNormalization())
CNN_model.add(Dropout(0.5))

# Classification layer
CNN_model.add(Dense(5, activation = 'sigmoid'))

checkpoint = ModelCheckpoint('CNN_checkpoint', save_best_only = True, monitor = 'val_loss', verbose = 1)

CNN_model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

history = CNN_model.fit_generator(training_set, validation_data = test_set, epochs = 5, verbose = 1, callbacks = [checkpoint])
```

```
# Creating the 5 Convolutional Layered model architecture
CNN_model = Sequential()

# First Layer
CNN_model.add(Conv2D(filters=32, kernel_size=3, padding = 'same', input_shape = [64, 64, 3], kernel_initializer = 'he_normal', ac
CNN_model.add(BatchNormalization())
CNN_model.add(MaxPooling2D(3))
CNN_model.add(Dropout(0.25))

# Second Layer
CNN_model.add(Conv2D(filters=64,kernel_size= 3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'))
CNN_model.add(BatchNormalization())

# Third Layer
CNN_model.add(Conv2D(filters=64,kernel_size= 3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'))
CNN_model.add(BatchNormalization())
CNN_model.add(MaxPooling2D())
CNN_model.add(Dropout(0.25))

# Fourth Layer
CNN_model.add(Conv2D(filters=128, kernel_size=3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'))
CNN_model.add(BatchNormalization())

# Fifth Layer
CNN_model.add(Conv2D(filters=128,kernel_size= 2, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'))
CNN_model.add(BatchNormalization())
CNN_model.add(MaxPooling2D())
CNN_model.add(Dropout(0.25))

# Fully Connected Layer
CNN_model.add(Flatten())
CNN_model.add(Dense(1024, activation = 'relu', kernel_initializer = 'he_normal'))
CNN_model.add(BatchNormalization())
CNN_model.add(Dropout(0.5))

# Classification layer
CNN_model.add(Dense(5, activation = 'sigmoid'))

checkpoint = ModelCheckpoint('CNN_checkpoint', save_best_only = True, monitor = 'val_loss', verbose = 1)

CNN_model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

history = CNN_model.fit_generator(training_set, validation_data = test_set, epochs = 10, verbose = 1, callbacks = [checkpoint])
```

```
## pant
image = cv.imread(r'manshort1.jfif')
resized = cv.resize(image, (input_shape[1], input_shape[0]))
resized = (resized / 255.0).reshape(-1, input_shape[1], input_shape[0], input_shape[2])
preds = CNN_model.predict(resized)
preds = zip(['dress','pants','shirt','shoes','shorts'], list(preds[0]))
preds = sorted(list(preds), key = lambda z: z[1], reverse = True)[:1][0]
plt.imshow(Image.open("manshort1.jfif"))
plt.show()
preds
```

```
# Dresses
image = cv.imread(r'dress.jpg')
resized = cv.resize(image, (input_shape[1], input_shape[0]))
resized = (resized / 255.0).reshape(-1, input_shape[1], input_shape[0], input_shape[2])
preds = CNN_model.predict(resized)
preds = zip(['dress','pants','shirt','shoes','shorts'], list(preds[0]))
preds = sorted(list(preds), key = lambda z: z[1], reverse = True)[:1][0]
plt.imshow(Image.open("dress.jpg"))
plt.show()
preds
```

```
# Reading the pixel values of all the images into "data_csv" dataframe, copying the dataframe to "data1" and removing the uneccess
data_csv = pd.read_csv("pixels.csv")
data1 = data_csv.copy()
data1 = data1.drop("Unnamed: 0",axis=1)
```

```
# Loading the categories into "Categories" list
Categories = data1["Category"]
```

```
# Dropping the Category column as we are not considering the labels for training the models.
data1 = data1.drop("Category",axis=1)
```

```
data3 = data1.copy()
```

```
# Applying the Normalization on the entire "data3" dataframe
data3 = data3/255
```

```
# Applying the StandardScaler function to transform the data
from sklearn.preprocessing import MinMaxScaler,StandardScaler
from sklearn.cluster import KMeans
scaler=StandardScaler()
scaler.fit(data3)
scaled_data=scaler.transform(data3)
```

```
# Reducing the dimensions from 900 to 2
pca=PCA(n_components=2)
pca.fit(scaled_data)
x_pca=pca.transform(scaled_data)
```

```
# Structure after StandardScaler and Principle component Analysis
print(scaled_data.shape)
print(x_pca.shape)
```

```
# Creating the Sum of Squared errors to plot Elbow curve
Sum_of_squared_errors = []
K = range(1,20)
for k in K:
    km = KMeans(n_clusters=k)
    km = km.fit(x_pca)
    Sum_of_squared_errors.append(km.inertia_)
```

```
# Plotting the Elbow curve
plt.plot(K, Sum_of_squared_errors, 'bx-')
plt.xlabel('K Values')
plt.ylabel('Sum_of_squared_errors')
plt.title('Elbow Method For Optimal k')
plt.show()
```

```
# Applying the k-means algorithm with 5 Cluster size
kmeansmodel = KMeans(n_clusters= 5, init='k-means++', random_state=42)
y_kmeans= kmeansmodel.fit_predict(x_pca)
cent = kmeansmodel.cluster_centers_
cluster_labels = kmeansmodel.labels_
```

```
# Representing the CLusters in graphical representation
plt.scatter(x_pca[y_kmeans==0,0],x_pca[y_kmeans==0,1],s=10, c='purple',label='Cluster1')
#plt.show()
plt.scatter(x_pca[y_kmeans==1,0],x_pca[y_kmeans==1,1],s=10, c='blue',label='Cluster2')
#plt.show()
plt.scatter(x_pca[y_kmeans==2,0],x_pca[y_kmeans==2,1],s=10, c='green',label='Cluster3')
#plt.show()
plt.scatter(x_pca[y_kmeans==3,0],x_pca[y_kmeans==3,1],s=10, c='cyan',label='Cluster4')
#plt.show()
plt.scatter(x_pca[y_kmeans==4,0],x_pca[y_kmeans==4,1],s=10, c='magenta',label='Cluster5')
#plt.show()
plt.scatter(kmeansmodel.cluster_centers_[:,0], kmeansmodel.cluster_centers_[:,1],s=50,marker='s', c='black', alpha=0.7, label='Ce
plt.legend()
plt.show()
```

```
# Calculating the accuracies of each Cluster and overall accuracy of the model
sample_silhouette_values = metrics.silhouette_samples(x_pca, cluster_labels)

means_lst = []
for label in range(5):
    means_lst.append(sample_silhouette_values[cluster_labels == label].mean())
print("Cluster 0: "+ str(means_lst[0]))
print("Cluster 1: "+ str(means_lst[1]))
print("Cluster 2: "+ str(means_lst[2]))
print("Cluster 3: "+ str(means_lst[3]))
print("Cluster 4: "+ str(means_lst[4]))
print("Average Silhouette Accuracy : %0.3f"% metrics.silhouette_score(x_pca, cluster_labels,metric='euclidean'))
```

```
data3["Predicted"]=y_kmeans
data3["Actual"]=Categories
```

```
data4 = data3[["Predicted","Actual","pixel1"]]
```

```
kmeans_interpret = data4.groupby(["Predicted","Actual"]).count().reset_index()
```

```
kmeans_interpret.columns =["Predicted","Actual","Number_of_Images"]
```

```
kmeans_interpret
```

```
# Creating the "kmeans_dataset" which consists all the images from the dataset
dataset = ImageDataGenerator(rescale = 1./255)
kmeans_dataset = dataset.flow_from_directory('CNN//',
                                             target_size = (224, 224),
                                             batch_size = 32,
                                             class_mode = 'categorical')
```

```
# Creating the 5 Convolutional Layered CNN model which is used to extract features from the images.
unsupervised_model = Sequential()

# First Layer
unsupervised_model.add(Conv2D(filters=32, kernel_size=3, padding = 'same', input_shape = [224, 224, 3], kernel_initializer = 'he_
unsupervised_model.add(BatchNormalization())
unsupervised_model.add(MaxPooling2D(3))
unsupervised_model.add(Dropout(0.25))
# Second Layer
unsupervised_model.add(Conv2D(filters=64,kernel_size= 3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu')
unsupervised_model.add(BatchNormalization())
# Third Layer
unsupervised_model.add(Conv2D(filters=64,kernel_size= 3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu')
unsupervised_model.add(BatchNormalization())
unsupervised_model.add(MaxPooling2D())
unsupervised_model.add(Dropout(0.25))
# Fourth Layer
unsupervised_model.add(Conv2D(filters=128, kernel_size=3, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'
unsupervised_model.add(BatchNormalization())
# Fifth Layer
unsupervised_model.add(Conv2D(filters=128,kernel_size= 2, padding = 'same', kernel_initializer = 'he_normal', activation = 'relu'
unsupervised_model.add(BatchNormalization())
unsupervised_model.add(MaxPooling2D())
unsupervised_model.add(Dropout(0.25))
# Flatten Layer to bring the multi-dimension features to one dimension
unsupervised_model.add(Flatten())
unsupervised_model.compile(optimizer = 'adam', loss = 'categorical_crossentropy')
```

```
# Applying the images to extract the features
predicted_features= unsupervised_model.predict(kmeans_dataset)
```

```
# Applying the StandardScaler function to transform the data
scaler=StandardScaler()
scaler.fit(predicted_features)
scaled_data=scaler.transform(predicted_features)
```

```
# Reducing the dimensions from 900 to 2
pca=PCA(n_components=2)
pca.fit(scaled_data)
x_pca=pca.transform(scaled_data)
```

```
# Applying the k-means algorithm with 5 Cluster size
kmeansmodel = KMeans(n_clusters= 5, init='k-means++', random_state=728)
y_kmeans= kmeansmodel.fit_predict(x_pca)
cent = kmeansmodel.cluster_centers_
cluster_labels = kmeansmodel.labels_
```

```
# Representing the Clusters in graphical representation
plt.scatter(x_pca[y_kmeans==0,0],x_pca[y_kmeans==0,1],s=10, c='purple',label='Cluster1')
plt.show()
plt.scatter(x_pca[y_kmeans==1,0],x_pca[y_kmeans==1,1],s=10, c='blue',label='Cluster2')
plt.show()
plt.scatter(x_pca[y_kmeans==2,0],x_pca[y_kmeans==2,1],s=10, c='green',label='Cluster3')
plt.show()
plt.scatter(x_pca[y_kmeans==3,0],x_pca[y_kmeans==3,1],s=10, c='cyan',label='Cluster4')
plt.show()
plt.scatter(x_pca[y_kmeans==4,0],x_pca[y_kmeans==4,1],s=10, c='magenta',label='Cluster5')
plt.show()
plt.scatter(kmeansmodel.cluster_centers_[:,0], kmeansmodel.cluster_centers_[:,1],s=50,marker='s', c='black', alpha=0.7, label='Ce
plt.legend()
plt.show()
```

```
# Calculating the accuracies of each Cluster and overall accuracy of the model
sample_silhouette_values = metrics.silhouette_samples(x_pca, cluster_labels)

means_lst = []
for label in range(5):
    means_lst.append(sample_silhouette_values[cluster_labels == label].mean())
print("Cluster 0: "+ str(means_lst[0]))
print("Cluster 1: "+ str(means_lst[1]))
print("Cluster 2: "+ str(means_lst[2]))
print("Cluster 3: "+ str(means_lst[3]))
print("Cluster 4: "+ str(means_lst[4]))
print("Average Silhouette Accuracy : %0.3f"% metrics.silhouette_score(x_pca, cluster_labels,metric='euclidean'))
```

```
# Loading the pre-trained VGG16 model with imagenet weights and exluding the Fully connected and Classification layers for transf
VGG_model = VGG16(weights='imagenet',include_top=False,input_shape=(224,224,3))
```

```
# Summary of VGG16
VGG_model.summary()
```

```
# Applying the images to extract the features from the images
predicted_VGGfeatures= VGG_model.predict(kmeans_dataset)
```

```
# taking a copy of the Features
to_reshape = predicted_VGGfeatures.copy()
```

```
# Reducing the 7x7x512 feature dimensions to 14822x25088 dataframe
images_temp = to_reshape.reshape(14822, -1)
```

```
# Applying the StandardScaler function to transform the data
scaler=StandardScaler()
scaler.fit(images_temp)
scaled_data=scaler.transform(images_temp)
```

```
# Reducing the dimensions from 900 to 2
pca=PCA(n_components=2)
pca.fit(scaled_data)
x_pca=pca.transform(scaled_data)
```

```
# Applying the k-means algorithm with 5 Cluster size
kmeansmodel = KMeans(n_clusters= 5, init='k-means++', random_state=728)
y_kmeans= kmeansmodel.fit_predict(x_pca)
cent = kmeansmodel.cluster_centers_
cluster_labels = kmeansmodel.labels_
```

```
# Representing the Clusters in graphical representation
plt.scatter(x_pca[y_kmeans==0,0],x_pca[y_kmeans==0,1],s=10, c='purple',label='Cluster1')
#plt.show()
plt.scatter(x_pca[y_kmeans==1,0],x_pca[y_kmeans==1,1],s=10, c='blue',label='Cluster2')
#plt.show()
plt.scatter(x_pca[y_kmeans==2,0],x_pca[y_kmeans==2,1],s=10, c='green',label='Cluster3')
#plt.show()
plt.scatter(x_pca[y_kmeans==3,0],x_pca[y_kmeans==3,1],s=10, c='cyan',label='Cluster4')
#plt.show()
plt.scatter(x_pca[y_kmeans==4,0],x_pca[y_kmeans==4,1],s=10, c='magenta',label='Cluster5')
#plt.show()
plt.scatter(kmeansmodel.cluster_centers_[:,0], kmeansmodel.cluster_centers_[:,1],s=50,marker='s', c='black', alpha=0.7, label='Ce
plt.legend()
plt.show()
```

```
# Calculating the accuracies of each Cluster and overall accuracy of the model
sample_silhouette_values = metrics.silhouette_samples(x_pca, cluster_labels)

means_lst = []
for label in range(5):
    means_lst.append(sample_silhouette_values[cluster_labels == label].mean())
print("CLuster 0: "+ str(means_lst[0]))
print("CLuster 1: "+ str(means_lst[1]))
print("CLuster 2: "+ str(means_lst[2]))
print("CLuster 3: "+ str(means_lst[3]))
print("CLuster 4: "+ str(means_lst[4]))
print("Average Silhouette Accuracy : %0.3f"% metrics.silhouette_score(x_pca, cluster_labels,metric='euclidean'))
```