A Mini Project Report on

FASHION RECOMMENDATION SYSTEM USING SOCIAL MEDIA WEBSITE

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

in partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

K.NIKHIL SARADHI	**21831A0587**
K.SRAVAN KUMAR	**21831A0590**
M.TEJESH	**21831A05B0**

Under the Esteemed Guidance of

Mrs. P.ANUSHA

Assistant Professor, GNIT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GURU NANAK INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUH-Hyderabad)

Ranga Reddy District-501506

2024-2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GURU NANAK INSTITUTE OF TECHNOLOGY
(Affiliated to JNTUH-Hyderabad)

Ranga Reddy District-501506

CERTIFICATE

This is to certify that the project entitled **" FASHION RECOMMENDATION SYSTEM USING SOCIAL MEDIA WEBSITE"** is being presented with a report by **K.NIKHIL SARADHI (21831A0587), K.SRAVAN KUMAR (21831A0590), M.TEJESH (21831A05B0)** in partial fulfillment for the award of **Degree of Bachelor of Technology in Computer Science and Engineering,** to **Jawaharlal Nehru Technological University, Hyderabad**.

Mrs. P.ANUSHA
Internal Guide

Mr. D. SRINIVAS
Project Coordinator

Dr. B. SANTHOSH KUMAR
Professor and Head of the Department

EXTERNAL EXAMINER

RAM Innovative Infotech
M : +91 9581 012 012
E : raminnovativeinfotech@gmail.com
Flat No.#309, Amrutha Ville,
Opp: Yashoda Hospital, Somajiguda,
Hyderabad-82, Telangana, India
Www.raminnovativeinfotech.webs.com

Reach Your Aim by Mastering

PROJECT COMPLETION CERTIFICATE

This is to certify that the following students of final year B. Tech, Department of Computer Science Engineering - Guru Nanak Institute of Technology (GNIT) have completed their training and project at GNIT successfully.

STUDENT NAME:	Roll No:
1. K.Nikhil Saradhi	21831A0587
2. K.Sravan Kumar	21831A0590
3. M.Tejesh	21831A05B0

The training was conducted on Machine Learning - Deep Learning Technology for the completion of the project titled Fashion Recommendation System Using Social Media Website in 2024 - 2025. The project has been completed in all aspects.

Signature

GURU NANAK INSTITUTE OF TECHNOLOGY

Ibrahimpatnam, R.R. Dist. 501506.

VISION OF GNIT

- To be a world-class educational and research institution in the service of humanity by promoting high quality Engineering and Management Education.

MISSION OF GNIT

- Imbibe soft skills and technical skills.
- Develop the faculty to reach international standards.
- Maintain high academic standards and teaching quality that promotes the typical thinking and independent judgment.
- Promote research, innovation and Product development by collaboration with reputed foreign universities.
- Offer collaborative industry programs in emerging areas and spirit of enterprise.

GURU NANAK INSTITUTE OF TECHNOLOGY

Ibrahimpatnam, R.R. Dist. 501506.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION OF DEPARTMENT

- To be recognized as a leading department of Computer Science and Engineering in the region by students, employers and be known for leadership, Ethics, and commitment to fostering quality teaching-learning, research, and innovation.

MISSION OF DEPARTMENT

- Nurture young individuals into knowledgeable, skill-full and ethical professionals in their pursuit of Computer Science and Engineering.
- Nurture the faculty to expose them to world-class infrastructure.
- Sustain high performance by excellence in teaching, research and innovations.
- Extensive partnerships and collaborations with foreign universities for technology upgradation
- Develop Industry-Interaction for innovation and product development

GURU NANAK INSTITUTE OF TECHNOLOGY

Ibrahimpatnam, R.R. Dist. 501506.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES (PEO'S)

PEO-1: Graduates shall have the ability to apply knowledge and technical skills in emerging areas of Computer Science and Engineering for higher studies, research, employability, product development and handle realistic problems.

PEO-2: Graduates shall maintain ethical conduct, sense of responsibility to serve the society and protect the environment.

PEO-3: Graduates shall possess academic excellence with innovative insight, soft skills, managerial skills, leadership qualities, knowledge of contemporary issues for successful professional career.

PROGRAM OUTCOMES (PO'S)

PO-1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.

PO-4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO-6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-7: Environment and sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO'S)

PSO-1: Professional Skills: The ability to understand the principles and working of computer systems. Students can assess the hardware and software aspects of computer systems.

PSO-2: Problem Solving Skills: The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

PSO-3: Successful Career and Entrepreneurship: The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur ,and a zest for higher studies.

DECLARATION

We hereby declare that the Minor Project report entitled **"FASHION RECOMMENDATION SYSTEM USING SOCIAL MEDIA WEBSITE"** is the work done by **K.NIKHILSARADHI, K.SRAVAN KUMAR, M.TEJESH** bearing the roll no's **21831A0587, 21831A0590, 21831A05B0** towards the fulfillment of the requirement for the award of the **Degree of Bachelor of Technology in Computer Science and Engineering**, to **Jawaharlal Nehru Technological University, Hyderabad,** is the result of the work carried out under the guidance of **Mrs. P.ANUSHA**, Assistant Professor **Guru Nanak Institute of Technology, Hyderabad.** We further declare that this project report has not been previously submitted either in partor full for the award of any degree or diploma by any organization or university.

K.NIKHIL SARADHI 21831A0587

K.SRAVAN KUMAR 21831A0590

M.TEJESH 21831A05B0

ACKNOWLEDGEMENT

"Task successful" makes everyone happy. But the happiness will be gold without glitter if we didn't state the persons who have supported us to make it a success. We would like to express our sincere thanks and gratitude to our Principal, **Dr. K.VENKATA RAO,** and Head of the Department, **Dr. B. SANTHOSH KUMAR**, Department of Computer Science and Engineering, Guru Nanak Institute of Technology for having guided us in developing the requisite capabilities for taking up this project.

We thank Project Coordinator **Mr. D.SRINIVAS** Computer Science Engineering, Guru Nanak Institute of Technology for providing seamless support and right suggestions that are given in the development of the project.

We specially thank our internal guide **Mrs. P.ANUSHA**, Assistant Professor for her constant guidance in every stage of the project. We would also like to thank all our lecturers for helping us in every possible way whenever the need arose.

On a more personal note, we thank our beloved parents and friends for their moral support during the course of our project.

ABSTRACT

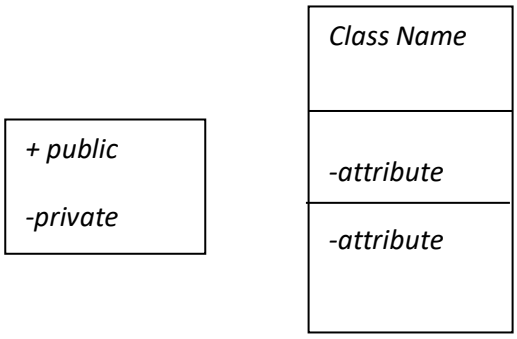
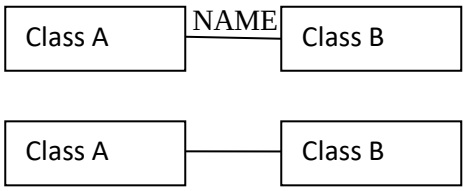
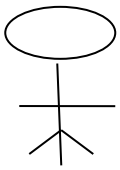
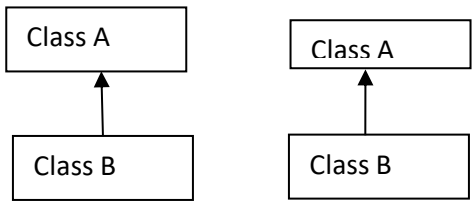
Fashion knowledge encourages people to properly dress and faces not only physiological necessity of users, but also the requirement of social practices and activities. It usually includes three jointly related aspects of: occasion, person and clothing. Nowadays, social media platforms allow users to interact with each other online to share opinions and information. The use of social media sites such as Instagram has already spread to almost every fashion brand and been evaluated as business take-off tools. With the heightened use of social media as a means of marketing communication for fashion brands, it has become necessary to empirically analyses and extract fashion knowledge from them.

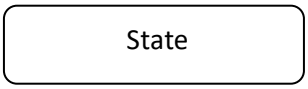
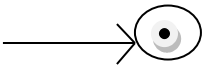
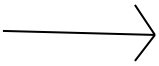
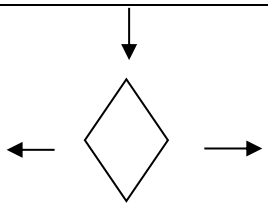
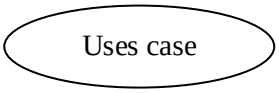
Thus, social brands are investing on them. In this way, they can understand the consumer's preferences. This change is also having a significant impact on social media data analysis. To solve this issue, the Deep learning (DL) methods are proven to be effective solutions due to their automatic learning capability. However, little systematic work currently exists on how researchers have applied DL for analyzing fashion knowledge from social media data. Hence, this contribution outlines DL-based techniques for social media data related to fashion domain. In this study, a review of the dataset within the fashion world and the DL methods applied on, it is presented to help out new researchers interested in this subject. In particular, two different tasks will be considered: Fashion Classification and Clothes Recommendation. Therefore, the purpose of this paper is to underline the multiple applications within the fashion world using deep learning techniques. However, this review does not cover all the methods used: in fact, only Deep Learning methods have been analyzed. This choice was made since, given the huge amount of fashion social media data that has been collected, Deep Learning methods achieve the best performance both in terms of accuracy and time. Limitations point towards unexplored areas for future investigations, serving as useful guidelines for future research directions.

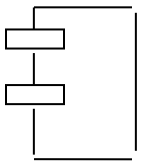
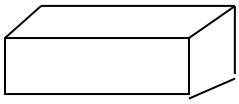
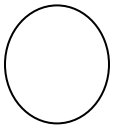
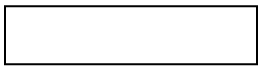
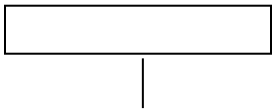
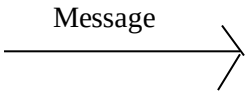
TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation		Interaction between the system and external environment

5.	Relation (uses)	uses	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the processes.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Use case		Interact ion between the system and external environment.

14.	Component		Represents physical modules which are a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message		Represents the message exchanged.

CHAPTER-1

INTRODUCTION

1.1 GENERAL

Online Social networks are part of every person's life. More than half of the world's population is connected to the internet and has at least one social platform. According to the report carried out by We Are Social of January 2021, in the world there are 7.83 billion people, 66.6% of these have a mobile phone. 4.66 billion People access the internet, an increase of 7.3% compared to January 2020. World internet penetration stands at 59.5%, but the values could be even higher by virtue of problems related to the correct tracking of internet users related to the COVID-19 pandemic. There are 4.20 billion users of social platforms, an increase of 13%. The use of social platforms therefore stands at 53% of the world population.

In particular, social networks have long since changed the way of communicating and perceiving the world: it is there-fore no coincidence that fashion, of which communication and perception are two fundamental pillars, is an integral part of this revolution. In fact, the fashion industry is one of the most dynamic in society and in this context social media are fundamental communication tools, in particular Facebook (born in 2004), Instagram (born in 2010) and TikTok (born in 2018).

Facebook was born in 2004 and, to date, is one of the most used social networks in the world, with over 2 billion active users. To date, many fashion brands are present on Facebook with a company page. The primary goal is to attract new customers and retain existing ones. A strategically managed Facebook page with careful publication of content will make a brand more attractive, involving an increasing number of users.

Instagram was born in 2010 and one of the strengths of this social network is the communicative power of the images that are able to convey the identity of a brand. TikTok was born in 2018 and it is a platform where users can express their creativity to the maximum through short videos between 15 and 60 seconds, with background music of all kinds.

The main social reference for the fashion domain is Instagram. However, leading fashion brands have proven the power of social media marketing across multiple channels. Each channel has different features to offer, giving new ways to achieve goals. Facebook is the most used

social media platform in the world with more of 2 billion monthly active users. In addition to regular Facebook posts, fashion app marketers can use the platform for live broadcasts.

1.2 LITERATURE SURVEY

Title: A review of digital fashion research: Before and beyond communication and marketing.

Author: T. H. Nobile, A. Noris, N. Kalbaska, and L. Cantoni.

Year: 2021

Description: This study focuses on the field of digital fashion and its development by providing an overview regarding fashion design and culture. It is part of a larger research that involved a literature review of 491 relevant papers. From the analysis of this corpus, three main categories were identified: Communication and Marketing, Design and Production and Culture and Society. This study focuses on the categories Design and Production and Culture and Society, which collectively gathered indicatively 48% of the selected literature. It presents its relevant studies and sub-categories, providing a rich and varied map of them and contributing to better design in further research in digital fashion.

Title: Multidisciplinary pattern recognition applications.

Author: M. Paolanti and E. Frontoni.

Year: 2020

Description: Pattern recognition (PR) is the study of how machines can examine the environment, learn to distinguish patterns of interest from their background, and make reliable and feasible decisions regarding the categories of the patterns. However, even after almost 70 years of research, the design of an application based on pattern recognizer remains an ambiguous goal. Moreover, currently, there are huge volumes of data that must be dealt with, which include image, video, text and web documents; DNA; microarray gene data; etc. Among the various frameworks in which pattern recognition has been traditionally formulated, the statistical and machine learning approaches have been most comprehensively studied and employed in practice. Recently, deep learning techniques and methods have been receiving increasing attention. The main objective of this review is to summarize PR applications, departing from the major algorithms used for their design. The PR approaches are subdivided into three main methods: machine learning, statistical, and deep learning. In order to evidence the multidisciplinary aspects of PR applications, attention has been focused on latest PR methods applied to five fields of research: biomedical and biology, retail, surveillance, social media intelligence, and digital cultural heritage. In this study, we discuss in detail the recent advances of PR approaches and propose the main applications within each field. We also present challenges and benchmarks in terms of advantages and disadvantages of the selected method in each field. A wide set of examples of applications in various domains are also provided, along with the specific method applied.

Title: A detailed review of artificial intelligence applied in the fashion and apparel industry.

Author: C. Giri, S. Jain, X. Zeng, and P. Bruniaux.

Year:2019

Description: The enormous impact of artificial intelligence has been realized in transforming the fashion and apparel industry in the past decades. However, the research in this domain is scattered and mainly focuses on one of the stages of the supply chain. Due to this, it is difficult to comprehend the work conducted in the distinct domain of the fashion and apparel industry. Therefore, this study aims to study the impact and the significance of artificial intelligence in the fashion and apparel industry in the last decades throughout the supply chain. Following this objective, we performed a systematic literature review of research articles (journal and conference) associated with artificial intelligence in the fashion and apparel industry. Articles were retrieved from two popular databases “Scopus” and “Web of Science” and the article screening was completed in five phases resulting in 149 articles. This was followed by article categorization which was grounded on the proposed taxonomy and was completed in two steps. First, the research articles were categorized according to the artificial intelligence methods applied such as machine learning, expert systems, decision support system, optimization, and image recognition and computer vision. Second, the articles were categorized based on supply chain stages targeted such as design, fabric production, apparel production, and distribution. In addition, the supply chain stages were further classified based on business-to-business (B2B) and business-to-consumer (B2C) to give a broader outlook of the industry. As a result of the categorizations, research gaps were identified in the applications of AI techniques, at the supply chain stages and from a business (B2B/B2C) perspective. Based on these gaps, the future prospects of the AI in this domain are discussed. These can benefit the researchers in academics and industrial practitioners working in the domain of the fashion and apparel industry.

Title: DeepFashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images.

Author: Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo.

Year:2019

Description: Understanding fashion images have been advanced by benchmarks with rich annotations such as DeepFashion, whose labels include clothing categories, landmarks, and consumer-commercial image pairs. However, DeepFashion has nonnegligible issues such as single clothing-item per image, sparse landmarks (4~8 only), and no per-pixel masks, making it had significant gap from real-world scenarios. We fill in the gap by presenting DeepFashion2 to address these issues. It is a versatile benchmark of four tasks including clothes detection, pose estimation, segmentation, and retrieval. It has 801K clothing items where each item has rich annotations such as style, scale, viewpoint, occlusion, bounding box, dense landmarks and masks. There are also 873K Commercial-Consumer clothes pairs. A strong baseline is proposed, called Match R-CNN, which builds upon Mask R-CNN to solve the above four tasks in an end-to-end manner. Extensive evaluations are conducted with different criterions in DeepFashion2.

Title: Understanding humans in crowded scenes: Deep nested adversarial learning and a new benchmark for multi-human parsing.

Author: J. Zhao, J. Li, Y. Cheng, T. Sim, S. Yan, and J. Feng.

Year:2018

Description: Despite the noticeable progress in perceptual tasks like detection, instance segmentation and human parsing, computers still perform unsatisfactorily on visually understanding humans in crowded scenes, such as group behavior analysis, person re-identification and autonomous driving, etc. To this end, models need to comprehensively perceive the semantic information and the differences between instances in a multi-human image, which is recently defined as the multi-human parsing task. In this study, we present a new large-scale database "Multi-Human Parsing (MHP)" for algorithm development and evaluation, and advances the state-of-the-art in understanding humans in crowded scenes. MHP contains 25,403 elaborately annotated images with 58 fine-grained semantic category labels, involving 2-26 persons per image and captured in real-world scenes from various viewpoints, poses, occlusion, interactions and background. We further propose a novel deep Nested Adversarial Network (NAN) model for multi-human parsing. NAN consists of three Generative Adversarial Network (GAN)-like sub-nets, respectively performing semantic saliency prediction, instance-agnostic parsing and instance-aware clustering. These sub-nets form a nested structure and are carefully designed to learn jointly in an end-to-end way. NAN consistently outperforms existing state-of-the-art solutions on our MHP and several other datasets, and serves as a strong baseline to drive the future research for multi-human parsing.

1.3 SCOPE OF THE PROJECT

To analyze the methods and techniques for each kind of fashion, summarize the main path and highlight their contributions.

1.4 OBJECTIVE

In the Fashion Classification, to make these applications effective. In particular, the difficulties caused by the clothing property must be considered:

1) Same clothing can be considered different depending on the point of view, and different clothing can be considered the same (the lower part of a dress that is particularly short can be classified as a skirt);
2) Clothing can be easily deformed by stretching or folding;
3) A picture of clothing can change; for example, the images can only contain the type of fabric, or models wearing a dress with that same fabric;
4) The images can be very different from each other, in the sense that they can have many different conditions, including different angles and lighting, cluttered backgrounds, and partially hidden by other objects or people;
5) Some classes of clothing have almost identical features and can be confused with each other. For example, the pants and tights classes are two classes that are very similar to each other and very difficult to distinguish;

Some clothing classes are very difficult to identify. For example, this may be due to their small size, such as accessories.

1.5 EXISTING SYSTEM

- We propose the Fashion Attributes Recognition Network (FAR Net) to simultaneously recognize three types of clothing attributes, including colors, categories, and patterns, in noisy-labeled images.
- FAR Net contains two main components built on top of a CNN image feature extractor. The first component is a Noise Correction Network extended based on the model.
- This network corrects noisy labels for images and generates corrected multi-labels of clothing colors and categories.
- The second component is a Pattern Classification Network that classifies each image into one of the five clothing patterns. We trained FAR Net using the MTL framework.

1.5.1 EXISTING SYSTEM DISADVANTAGES

- We describe our model architecture and the loss functions in this section.
- The Pattern Classification Network uses categorical cross-entropy loss.

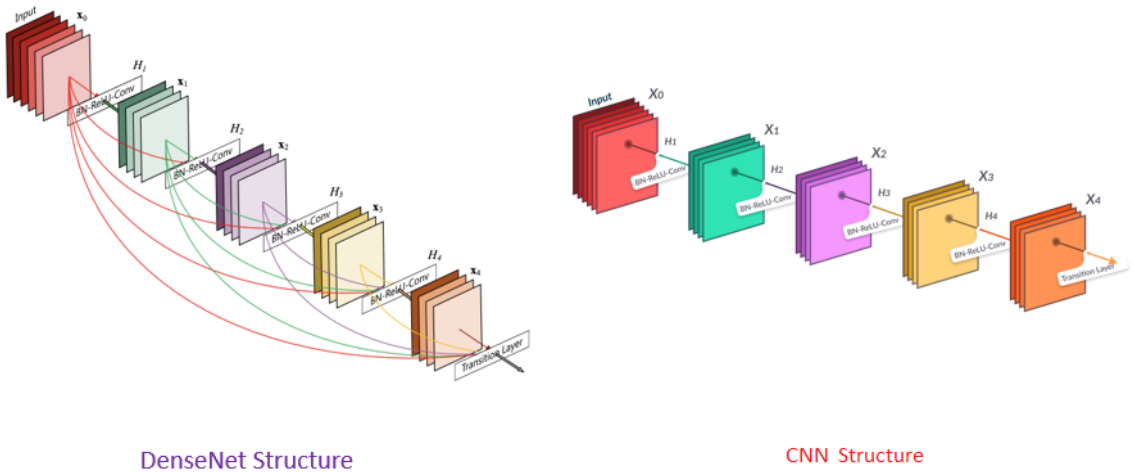
1.6 PROPOSED SYSTEM

- Researchers have proposed several fashion recommendation systems in the literature aiming at choosing the right outfit for different occasions.
- Companies therefore must try to analyze the information that is spontaneously generated by web users.
- Big Data analysis now makes it possible to predict future trends even before they explode, providing real-time information not only on the volume of sales, but also on that of online searches.
- More quickly identifying fabrics, styles and colors for which public interest is growing allows us to satisfy the request in a timely manner and consequently to sell more.

1.6.1 PROPOSED SYSTEM ADVANTAGES

- ML techniques, companies operating in the fashion sector can identify patterns in data and build models that can predict future results.
- This helps to create a more flexible and faster supply chain and manage inventory in an automated and intelligent way.
- To get more accurate results the keywords have been aggregated.

1.7 SYSTEM ARCHITECTURE



$$a^{[l]} = g\left([a^{[0]}, a^{[1]}, a^{[2]}, \ldots\ldots\ldots\ldots, a^{[l-1]}\right)$$

Fig 1.7: System Architecture Diagram

EXPLANATION:

The system architecture combines the DenseNet and CNN structures to enhance feature extraction and image classification efficiency. In the DenseNet structure, each layer is densely connected to all preceding layers, ensuring maximum feature reuse and mitigating the vanishing gradient issue by propagating information directly through shortcut connections. This design captures intricate patterns effectively and allows the network to learn robust representations. In contrast, the traditional CNN structure follows a sequential layer design where each layer outputs features to the next, focusing on progressively learning abstract representations. By leveraging DenseNet's dense connectivity and CNN's hierarchical learning, the architecture achieves improved accuracy in tasks such as fashion attribute recognition and recommendation, ensuring a robust and computationally efficient system.

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL

ML and DL techniques bring great benefits to image recognition and classification in the fashion environment. In fact, they can help to improve the user experience [147], which is a fundamental factor for the calculation of the Key Performance Indicator (KPI), which can be measured through factors such as the time spent by the user in front of the computer, the purchase volume and average checkout value.

Deep Learning methods, and in particular Convolutional Neural Networks, can help the user to have a more pleasant experience on the site, being able to make a quicker and more convenient search of the products. As a consequence, there will be an increase in KPIs, in the business profits and in the efficiency of the product management system. An online store that is multi-brand has to group the products and establish the rules necessary for unification and quality standard. When a brand shop proposes some products to the multi-brand online store, the manager reviews the incoming products and decides whether to approve or reject them. This methodology meets two different problems. The first problem is that paying the individual people who carry out the supervised learning process becomes a very expensive process. The second problem is that the time frame for carrying out this type of human reviews of different products in different stores is very long. One way to reduce costs and times, and consequently increase the performance and quality of the results, is the use of automatic systems based on CNN.

Considering the importance of clothing in society, there are many applications for Fashion Classification. An example is the prediction of clothing details in an image, that can help find similar clothing items in a dataset from e-commerce sites. Analogously, Fashion Classification based on user preferences can be used to provide recommendations to the user.

2.2 METHODOLOGIES

2.2.1 MODULES NAME

- **Dataset**
- **Importing the necessary libraries**
- **Retrieving the images**
- **Splitting the dataset**
- **Building the model**
- **Apply the model and plot the graphs for accuracy and loss**
- **Accuracy on test set**
- **Saving the Trained Model**

2.2.2 MODULES EXPLANATION

1) Dataset:

In the first module, we developed the system to get the input dataset for the training and testing purpose. We have taken the dataset for fashion classification and product recommendation.
The fashion classification dataset consists of 60000 fashion-mnist images
The product recommendation dataset consists of 5000 fashion-mnist images.

2) Importing the necessary libraries:

We will be using Python language for this. First we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, numpy ,matplotlib and tensorflow.

3) Retrieving the images:

We will retrieve the images and their labels. Then resize the images to (200,200) as all images should have same size for recognition. Then convert the images into numpy array.

4) Splitting the dataset:

Split the dataset into train and test. 80% train data and 20% test data.

A. Convolutional Neural Networks

The objectives behind the first module of the course 4 are:

- To understand the convolution operation
- To understand the pooling operation
- Remembering the vocabulary used in convolutional neural networks (padding, stride, filter, etc.)
- Building a convolutional neural network for multi-class classification in images

B. Computer Vision

Some of the computer vision problems which we will be solving in this article are:

1. Image classification
2. Object detection
3. Neural style transfer

One major problem with computer vision problems is that the input data can get really big. Suppose an image is of the size 68 X 68 X 3. The input feature dimension then becomes 12,288. This will be even bigger if we have larger images (say, of size 720 X 720 X 3). Now, if we pass such a big input to a neural network, the number of parameters will swell up to a HUGE number (depending on the number of hidden layers and hidden units). This will result in more computational and memory requirements – not something most of us can deal with

5) Building the model:

For building the model we will use sequential model from keras library. Then we will use CNN Model for fashion classification dataset and the DenseNet121 CNN Model for product recommendation dataset which consist of Convolutional layer with 64 filters and a 7x7 kernel size, with stride of 2 and padding of 3.

Batch normalization layer and ReLU activation layer.

Max pooling layer with a 3x3 kernel size and a stride of 2.

4 dense blocks, each containing several layers as follows:

A batch normalization layer and ReLU activation layer

A convolutional layer with 4k filters and a 1x1 kernel size
A batch normalization layer and ReLU activation layer
A convolutional layer with 32 filters and a 3x3 kernel size
Concatenation of the input with the output of the convolutional layer
Transition layers between the dense blocks, consisting of:
A batch normalization layer and ReLU activation layer
A convolutional layer with 4k filters and a 1x1 kernel size
A max pooling layer with a 2x2 kernel size and a stride of 2
Final dense layer for classification with 1000 output classes

6) Apply the model and plot the graphs for accuracy and loss:

We will compile the model and apply it using fit function. Then we will plot the graphs for accuracy and loss.

7) Accuracy on training set:

We got an accuracy of 99.2% on training set.

8) Saving the Trained Model:

Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle .
Make sure you have pickle installed in your environment.
Next, let's import the module and dump the model into.pkl file

2.3 TECHNIQUE USED OR ALGORITHM USED

2.3.1 EXISTING TECHNIQUE USED OR ALGORITHM USED

- **Clustering Algorithm:**

The second column lists the number of clusters for each clustering algorithm.

The number of clusters derived from DBSCAN is 45, which is the highest among all clustering algorithms**.**

The number of clusters derived from DBSCAN is 45, which is the highest among all clustering algorithms**.**

We clustered the street fashion images in each subset of Rich Wear by year in order to discover fashion trends**.**

2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED

- **CNN & DENSENET121:**

ML and DL techniques bring great benefits to image recognition and classification in the fashion environment.

In fact, they can help to improve the user experience, which is a fundamental factor for the calculation of the Key Performance Indicator (KPI), which can be measured through factors such as the time spent by the user in front of the computer, the purchase volume and average checkout value.

Deep Learning methods, and in particular Convolutional Neural Networks, can help the user to have a more pleasant experience on the site, being able to make a quicker and more convenient search of the products.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 GENERAL

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 4GB DD RAM
- HARD DISK : 250 GB

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating System : Windows 7/8/10
- Platform : Spyder3
- Programming Language : Python
- Front End : Spyder3

3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior,Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based duplications systems by resorting to the hybrid cloud architecture.

3.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

Usability

The system is designed with completely automated process hence there is no or less user intervention.

Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

Performance

This system is developing in the high-level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

Implementation

The system is implemented in web environment using Jupyter notebook software. The server is used as the intelligence server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

CHAPTER 4

SYSTEM DESIGN

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

4.2 UML DIAGRAMS

4.2.1 USE CASE DIAGRAM

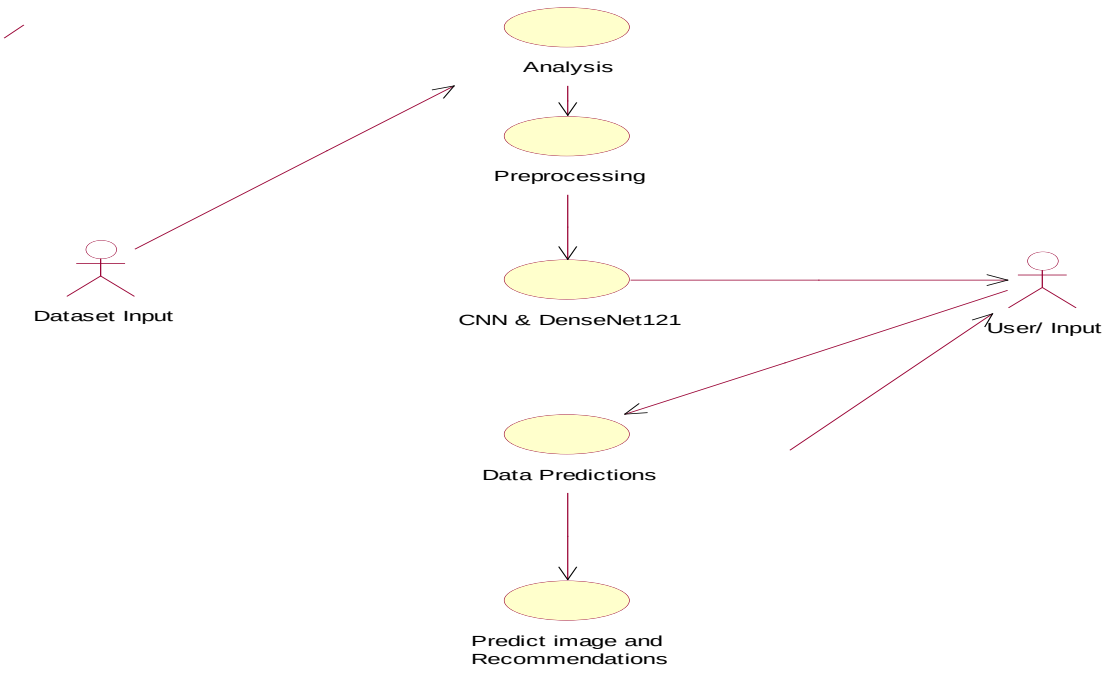


Fig 4.2.1: Use Case Diagram

EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

4.2.2 CLASS DIAGRAM

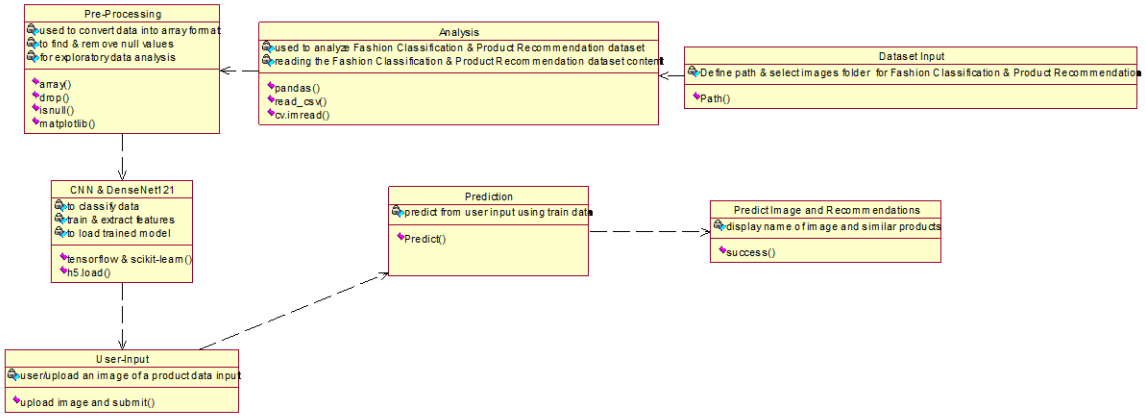


Fig 4.2.2: Class Diagram

EXPLANATION:

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

4.2.3 OBJECT DIAGRAM

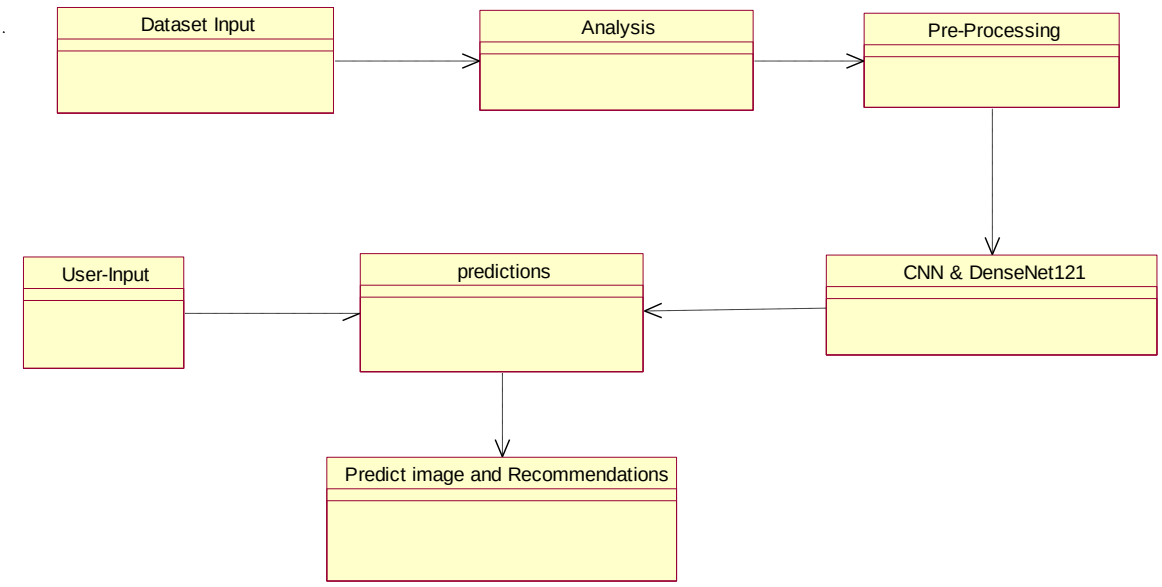


Fig 4.2.3: Object Diagram

EXPLANATION:

In the above diagram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

4.2.4 STATE DIAGRAM

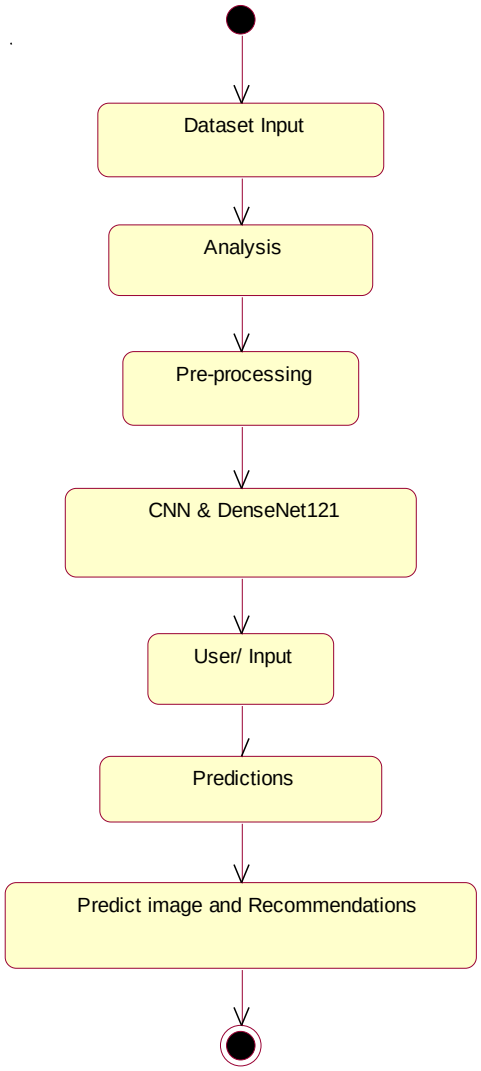


Fig 4.2.4: State Diagram

EXPLANATION:

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

4.2.5 ACTIVITY DIAGRAM

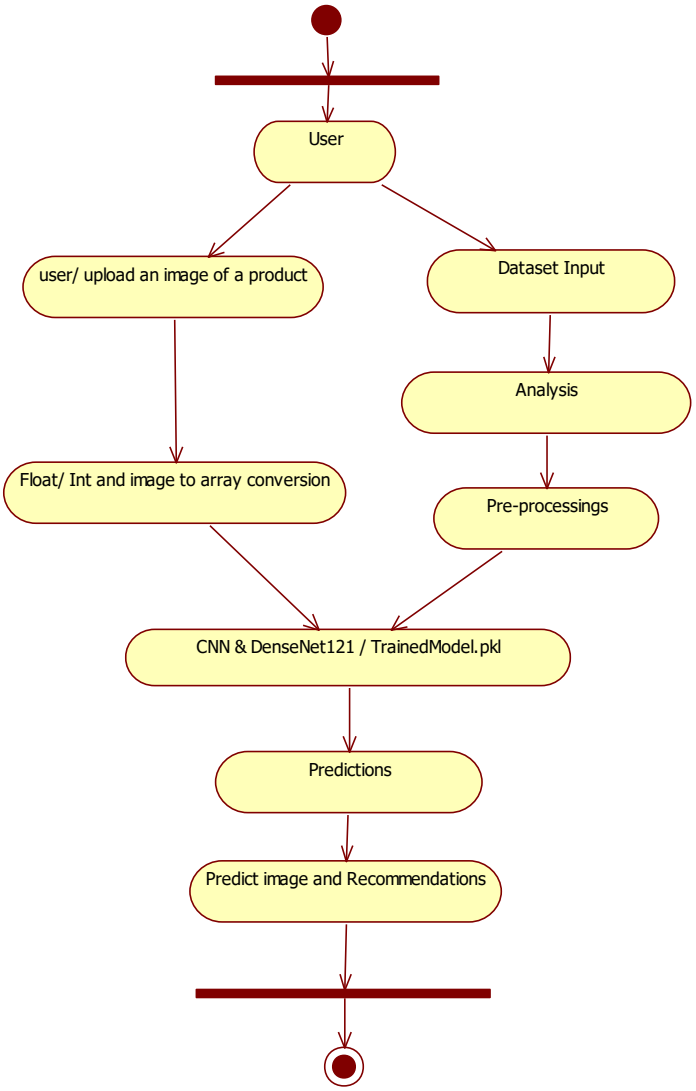


Fig 4.2.5: Activity Diagram

EXPLANATION:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

4.2.6 SEQUENCE DIAGRAM

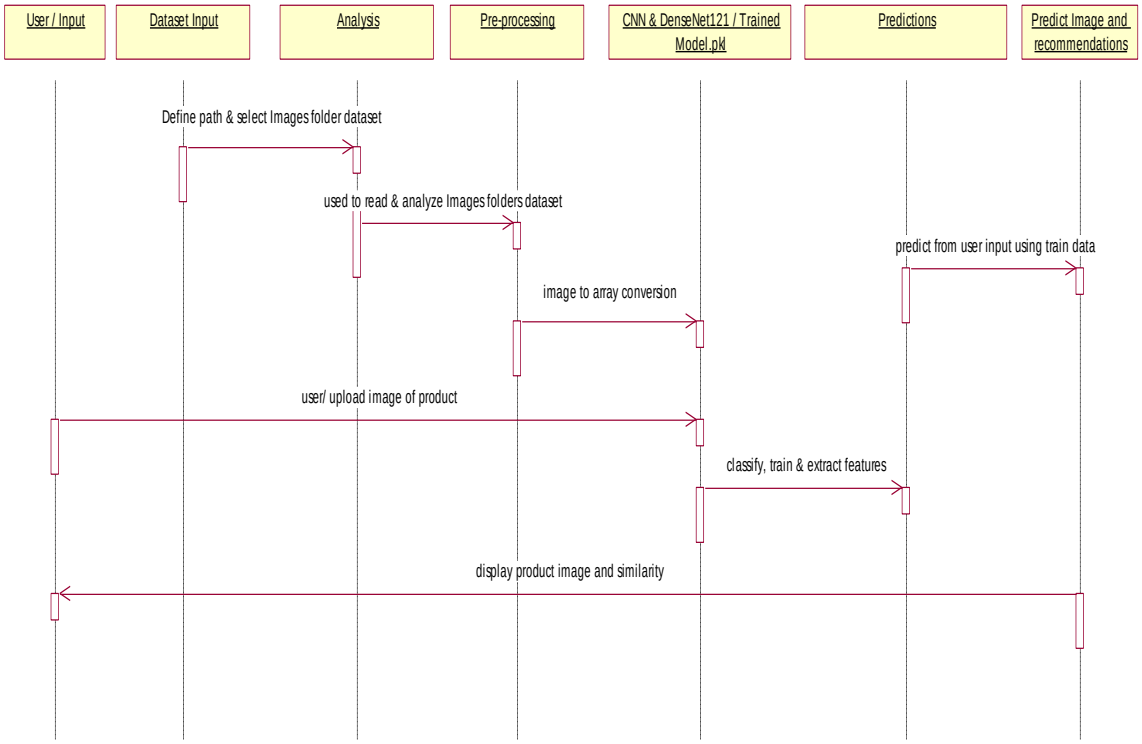


Fig 4.2.6: Sequence Diagram

EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

4.2.7 COLLABORATION DIAGRAM

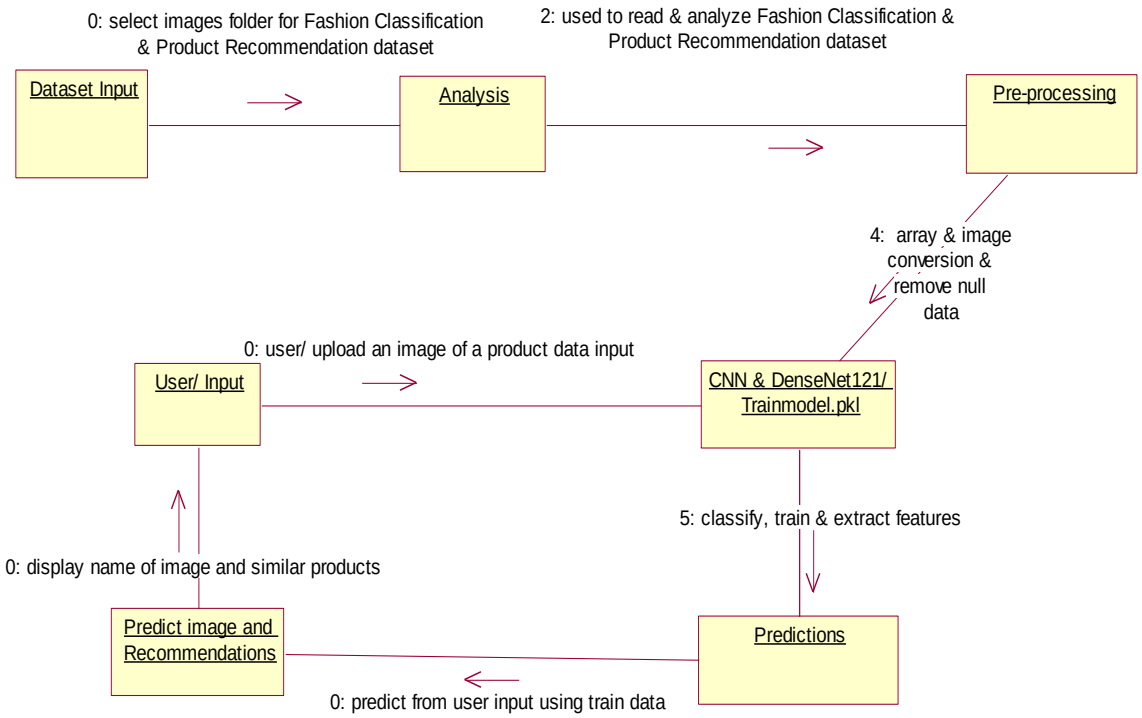


Fig 4.2.7: Collaboration Diagram

EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

4.2.8 COMPONENT DIAGRAM

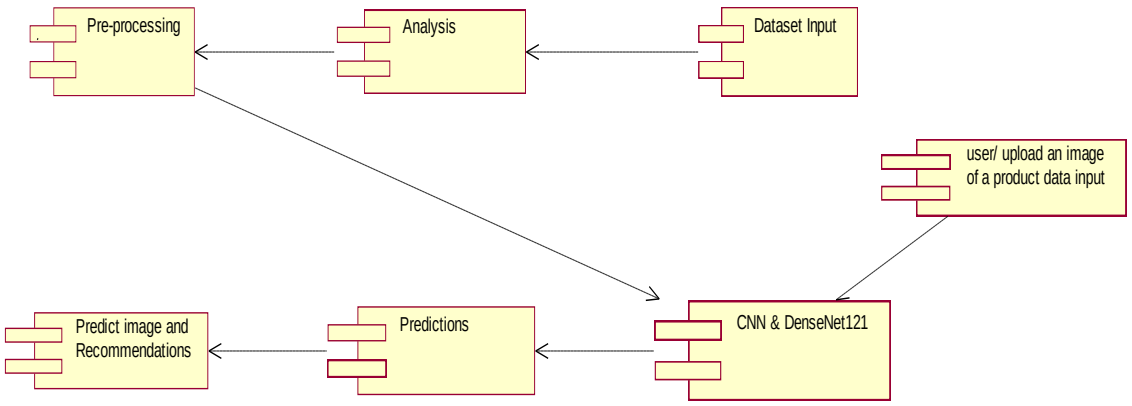


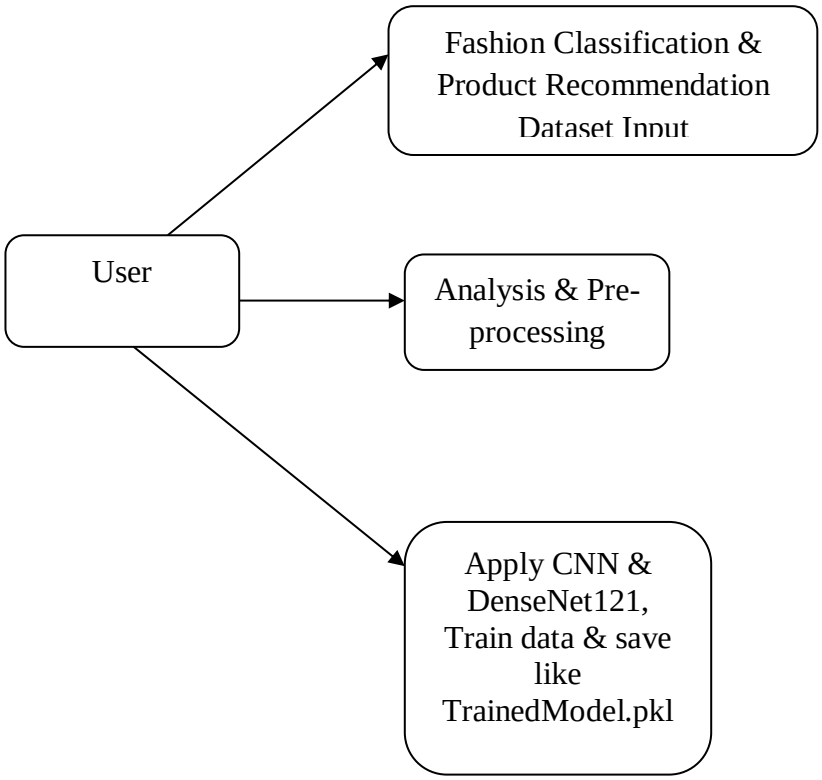
Fig 4.2.8: Component Diagram

EXPLANATION:

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

4.2.9 DATA FLOW DIAGRAM

Level 0



Level 1

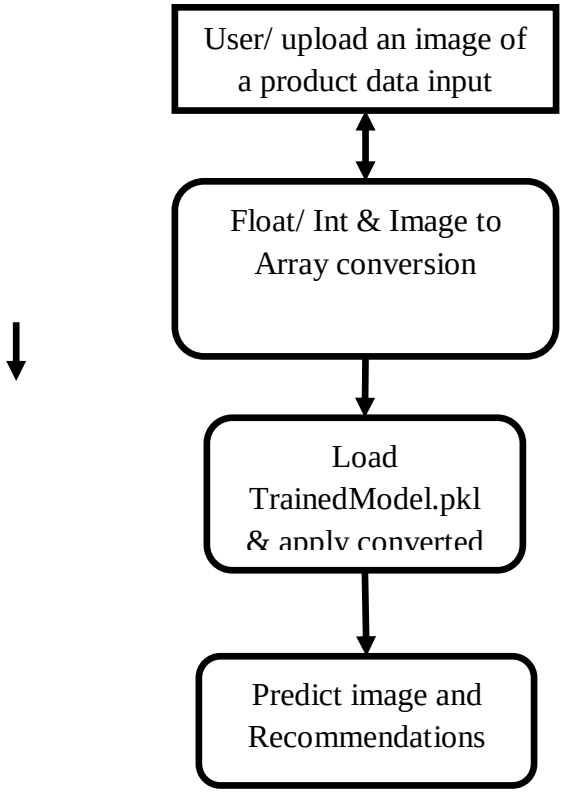


Fig 4.2.9 : Data Flow Diagram

EXPLANATION:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

4.2.10 DEPLOYMENT DIAGRAM

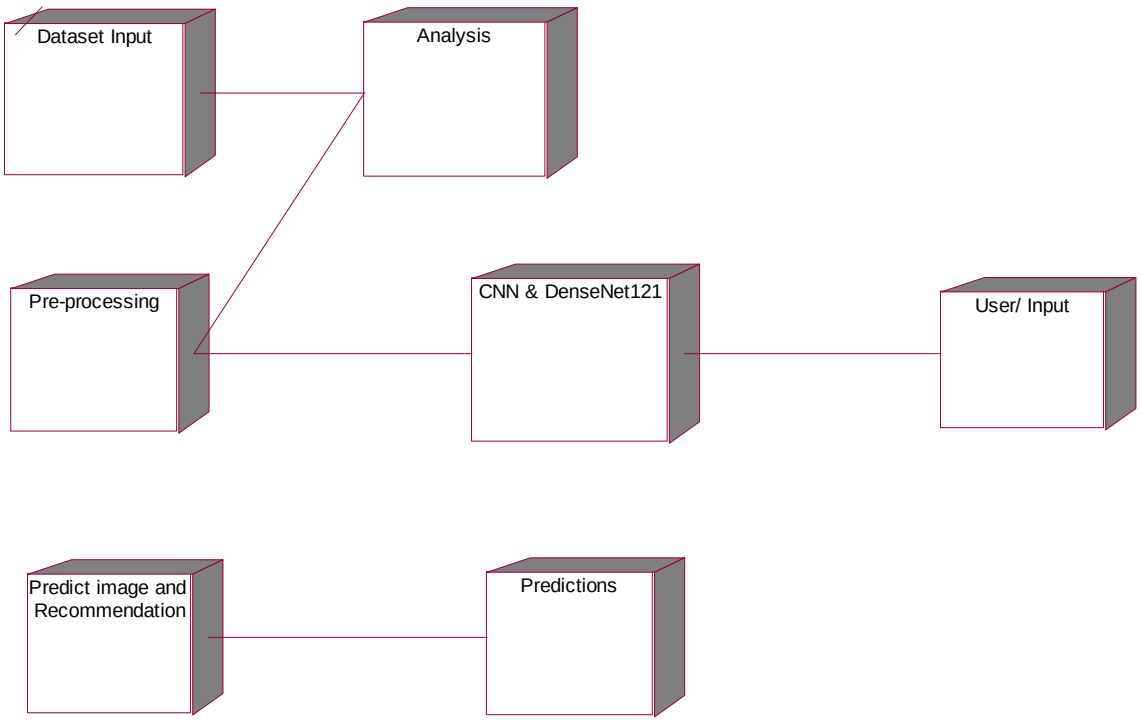


Fig 4.2.10: Deployment Diagram

EXPLANATION:

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

-

CHAPTER 5

DEVELOPMENT TOOLS

5.1 GENERAL

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

5.2 HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

5.3 IMPORTANCE OF PYTHON

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

5.4 FEATURES OF PYTHON

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

5.5 LIBRARIES OF PYTHON

- **numpy** - mainly useful for its N-dimensional array objects.
- **pandas** - Python data analysis library, including structures such as dataframes.
- **matplotlib** - 2D plotting library producing publication quality figures.
- **scikit-learn** - the machine learning algorithms used for data analysis and data mining tasks.

Figure : NumPy, Pandas, Matplotlib, Scikit-learn

CHAPTER 6

IMPLEMENTATION

6.1 GENERAL

Coding:

1. main.py

```
import streamlit as st
import os
from PIL import Image
import numpy as np
import pickle
import tensorflow
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50,preprocess_input
from sklearn.neighbors import NearestNeighbors
from numpy.linalg import norm
feature_list = np.array(pickle.load(open('embeddings.pkl','rb')))
filenames = pickle.load(open('filenames.pkl','rb'))
model = ResNet50(weights='imagenet',include_top=False,input_shape=(224,224,3))
model.trainable = False
model = tensorflow.keras.Sequential([
    model,
    GlobalMaxPooling2D()
])
st.title('Fashion Recommender System')
def save_uploaded_file(uploaded_file):
    try:
        with open(os.path.join('uploads',uploaded_file.name),'wb') as f:
            f.write(uploaded_file.getbuffer())
        return 1
```

```
    except:
        return 0
def feature_extraction(img_path,model):
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    expanded_img_array = np.expand_dims(img_array, axis=0)
    preprocessed_img = preprocess_input(expanded_img_array)
    result = model.predict(preprocessed_img).flatten()
    normalized_result = result / norm(result)
    return normalized_result
def recommend(features,feature_list):
    neighbors = NearestNeighbors(n_neighbors=6, algorithm='brute', metric='euclidean')
    neighbors.fit(feature_list)
    distances, indices = neighbors.kneighbors([features])
    return indices
# steps
# file upload -> save
uploaded_file = st.file_uploader("Choose an image")
if uploaded_file is not None:
    if save_uploaded_file(uploaded_file):
        # display the file
        display_image = Image.open(uploaded_file)
        st.image(display_image)
        # feature extract
        features = feature_extraction(os.path.join("uploads",uploaded_file.name),model)
        #st.text(features)
        # recommendention
        indices = recommend(features,feature_list)
        # show
        col1,col2,col3,col4,col5 = st.beta_columns(5)
        with col1:
```

```
            st.image(filenames[indices[0][0]])
        with col2:
            st.image(filenames[indices[0][1]])
        with col3:
            st.image(filenames[indices[0][2]])
        with col4:
            st.image(filenames[indices[0][3]])
        with col5:
            st.image(filenames[indices[0][4]])
    else:
        st.header("Some error occured in file upload")
```

2. run.py

```
from flask import Flask, render_template,request,redirect,url_for
import os
from werkzeug.utils import secure_filename
import pickle
import numpy as np
import tensorflow as tf
from numpy.linalg import norm
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from sklearn.neighbors import NearestNeighbors
import cv2
import instaloader
import mysql
import mysql.connector
app = Flask(__name__)

# Create an instance of the Instaloader class
```

```
loader = instaloader.Instaloader(dirname_pattern='downloads/{target}/{profile}')
#mysql connection
sql=mysql.connector.connect(
   host="localhost",
   user="root",
   password="root",
   database="instadata"
   )
cur = sql.cursor()
#cur.execute("CREATE TABLE register (Username varchar(30), phone bigint, email varchar(30), password varchar(20))")
#cur.close()

#main page
@app.route('/')
def home():
   return render_template('index.html')

# User Register
@app.route('/register', methods=["POST", "GET"])
def register():
   if request.method == "POST":
      username = request.form["text"]
      useremail = request.form["email"]
      password = request.form["password"]
      phn = request.form["number"]
         cur.execute("INSERT INTO register VALUES (%s, %s, %s, %s)",(username, phn, useremail, password))
      sql.commit()
      return render_template('contact.html')
   return render_template('contact.html')
```

```
#User Login
@app.route('/login', methods=["POST", "GET"])
def login():
    if request.method == "POST":
        useremail = request.form["email"]
        userpass = request.form["password"]
        #cur = sql.cursor()
            cur.execute('SELECT * FROM register WHERE email = %s AND password = %s',
(useremail, userpass))
        user = cur.fetchone()
        if user:
            return render_template("blog.html")
        else:
            return"Invalid credentials or user not found"
    return render_template('login.html')
############## About Page ######################################
@app.route('/about')
def about():
    return render_template('about.html')

# serch Page
@app.route('/search')
def search():
    return render_template('blog.html',similar_images=None)
#-------------------------------------------------------------------------#
UPLOAD_FOLDER = 'uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
STATIC_FOLDER = 'static'
app.config['STATIC_FOLDER'] = STATIC_FOLDER
@app.route('/search', methods=["GET", "POST"])
def search_user():
```

```
    similar_image_filenames = None
    if request.method == "POST":
        # Check if the POST request has the file part
        if 'image' not in request.files:
            return"No file part"
        file = request.files['image']
        # If the user does not select a file, the browser submits an empty file without a filename
        if file.filename == '':
            return"No selected file"
        if file:
            # Save the uploaded file to the UPLOAD_FOLDER
            filename = secure_filename(file.filename)
            file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
            file.save(file_path)

            # Process the uploaded image
            similar_image_filenames = process_uploaded_image(file_path)
    # Render the blog.html template with or without similar images
    return render_template('blog.html', similar_images=similar_image_filenames)
def process_uploaded_image(file_path):
    # Load precomputed features and filenames
    feature_list = np.array(pickle.load(open('savemodels.pkl','rb')))
    filenames = pickle.load(open('filenames.pkl','rb'))
    model = ResNet50(weights='imagenet', include_top=False, input_shape=(224,224,3))
    model.trainable = False
    model = tf.keras.Sequential([
        model,
        GlobalMaxPooling2D()
    ])
    img = image.load_img(file_path, target_size=(224,224))
    img_array = image.img_to_array(img)
```

```
    expanded_img_array = np.expand_dims(img_array, axis=0)
    preprocessed_img = preprocess_input(expanded_img_array)
    result = model.predict(preprocessed_img).flatten()
    normalized_result = result / np.linalg.norm(result)
    neighbors = NearestNeighbors(n_neighbors=7, algorithm='brute', metric='euclidean')
    neighbors.fit(feature_list)
    distances, indices = neighbors.kneighbors([normalized_result])
    similar_image_filenames = []
    for file_idx in indices[0][1:7]:
        # Get the filename relative to the static directory
        filename = os.path.relpath(filenames[file_idx], 'static')
        similar_image_filenames.append(filename)
    return similar_image_filenames
#service page
@app.route('/service')
def service():
    return render_template('service.html')
####################INSTA PAGE ######################################
@app.route('/instaprofile', methods=['GET', 'POST'])
def instaprofile():
    if request.method == 'POST':
        username = request.form['username']
        try:
            # Download the profile picture of the user
            loader.download_profile(username, profile_pic_only=True)
            return f"Profile picture for {username} downloaded successfully!"
        except Exception as e:
            return f"Error: {str(e)}"
    return render_template('instaprofile.html')
if __name__ == '__main__':
app.run()
```

3. test.py

```
import pickle
import tensorflow
import numpy as np
from numpy.linalg import norm
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50,preprocess_input
from sklearn.neighbors import NearestNeighbors
import cv2
feature_list = np.array(pickle.load(open('savemodels.pkl','rb')))
filenames = pickle.load(open('filenames.pkl','rb'))
model = ResNet50(weights='imagenet',include_top=False,input_shape=(224,224,3))
model.trainable = False
model = tensorflow.keras.Sequential([
    model,
    GlobalMaxPooling2D()
])

img = image.load_img('sample/shoes.jpg',target_size=(224,224))
img_array = image.img_to_array(img)
expanded_img_array = np.expand_dims(img_array, axis=0)
preprocessed_img = preprocess_input(expanded_img_array)
result = model.predict(preprocessed_img).flatten()
normalized_result = result / norm(result)

neighbors = NearestNeighbors(n_neighbors=6,algorithm='brute',metric='euclidean')
neighbors.fit(feature_list)
distances,indices = neighbors.kneighbors([normalized_result])
print(indices)
for file in indices[0][1:6]:
```

```
    temp_img = cv2.imread(filenames[file])
    cv2.imshow('output',cv2.resize(temp_img,(512,512)))
    cv2.waitKey(0)
```

4. app.py

```
import pickle
import tensorflow
import numpy as np
from numpy.linalg import norm
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50,preprocess_input
from sklearn.neighbors import NearestNeighbors
import cv2
feature_list = np.array(pickle.load(open('savemodels.pkl','rb')))
filenames = pickle.load(open('filenames.pkl','rb'))
model = ResNet50(weights='imagenet',include_top=False,input_shape=(224,224,3))
model.trainable = False
model = tensorflow.keras.Sequential([
    model,
    GlobalMaxPooling2D()
])

img = image.load_img('sample/shoes.jpg',target_size=(224,224))
img_array = image.img_to_array(img)
expanded_img_array = np.expand_dims(img_array, axis=0)
preprocessed_img = preprocess_input(expanded_img_array)
result = model.predict(preprocessed_img).flatten()
normalized_result = result / norm(result)
neighbors = NearestNeighbors(n_neighbors=6,algorithm='brute',metric='euclidean')
neighbors.fit(feature_list)
```

```
distances,indices = neighbors.kneighbors([normalized_result])
print(indices)
for file in indices[0][1:6]:
    temp_img = cv2.imread(filenames[file])
    cv2.imshow('output',cv2.resize(temp_img,(512,512)))
    cv2.waitKey(0)
```

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 GENERAL

This project is implements like application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

7.2 RESULTS AND DISCUSSIONS

Home page:

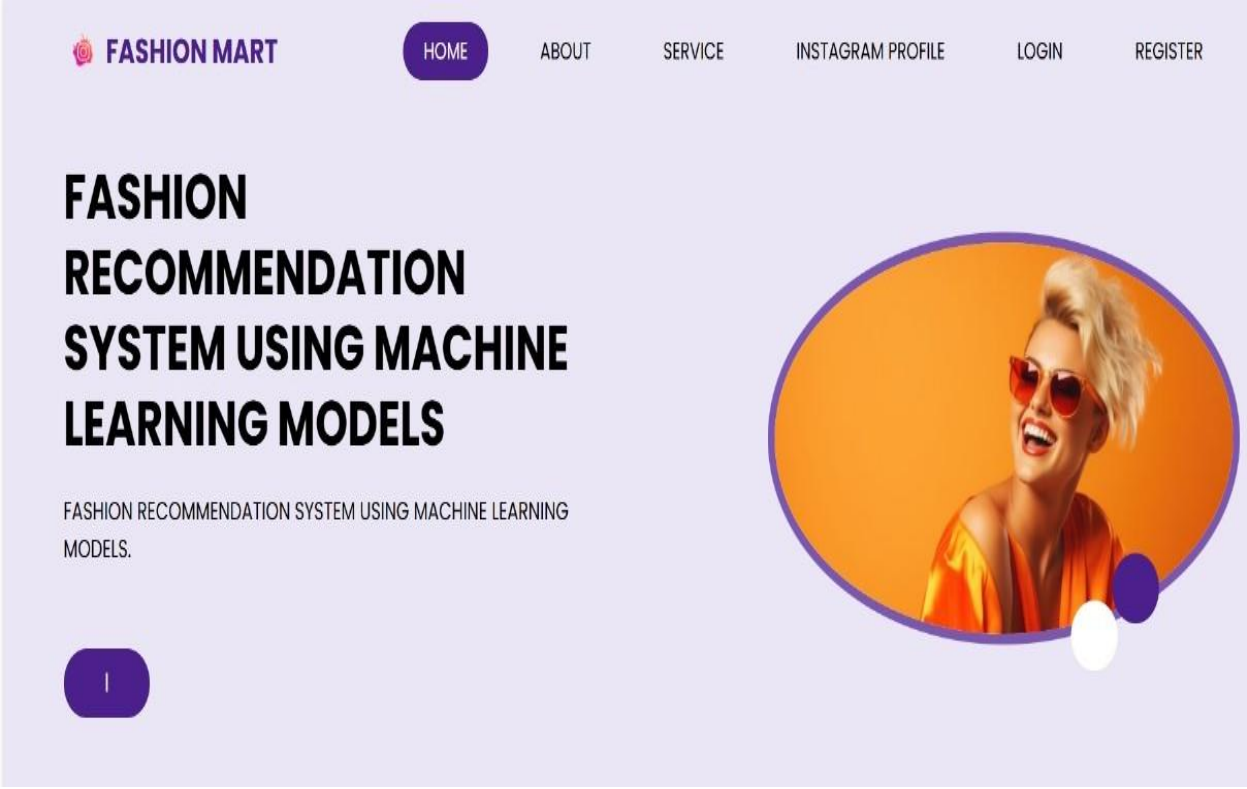


Fig: 7.2.1 Home Page

The home page is the main landing page of a website, serving as the gateway for users to exploreitscontentandfeatures.Ittypicallyincludesawelcomingmessageandaclean,visually appealing design that reflects the brand identity.

Registration Page:

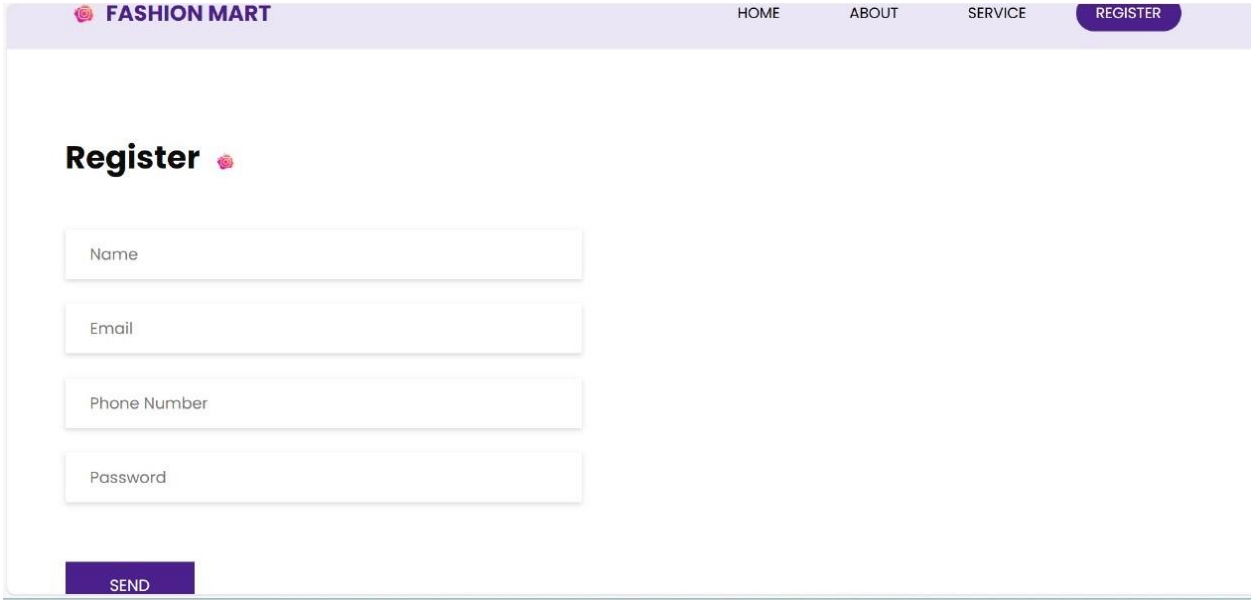


Fig: 7.2.2 Registration Page

The registration page is a user interface designed to facilitate the process of creating a new account. It typically features input fields for gathering essential information from the user, such as their name, email address, and a secure password.

Login Page:

Fig: 7.2.3 login Page

The login page is a user interface that allows users to securely access their accounts. It typically includes input fields for entering a username and password, as well as a button to submit the login credentials. The page may also include options for recovering a forgotten password or creating a new account.

Instagram page:

Fig: 7.2.4 Instagram Page

This webpage is a simple Instagram profile picture downloader. Users enter an Instagram username into a text field. Clicking "Download Profile Picture" presumably retrieves and downloads the corresponding profile image. Which can be further used for giving input for recommendation system

Input Page:

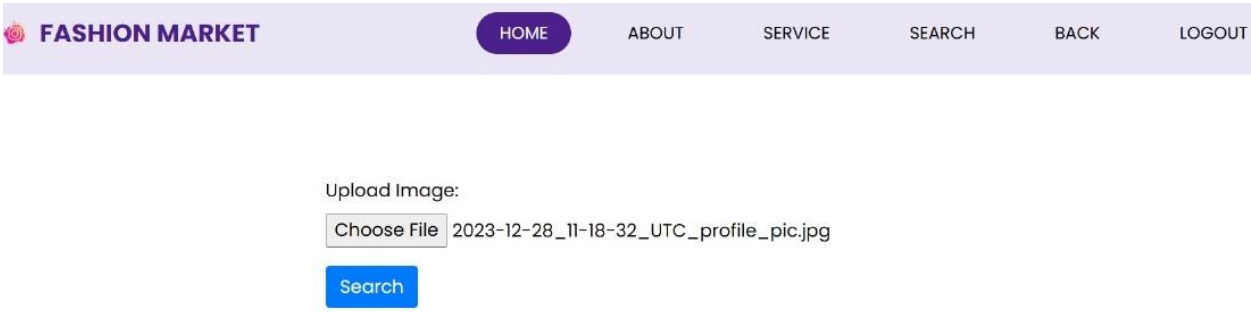


Fig: 7.2.5 Input Page

In the input page, user is asked to upload the image downloaded using the Instagram profile picture downloader and the model predicts the fashion of the user using the provided Instagram profile picture and gives recommendations

Result page:

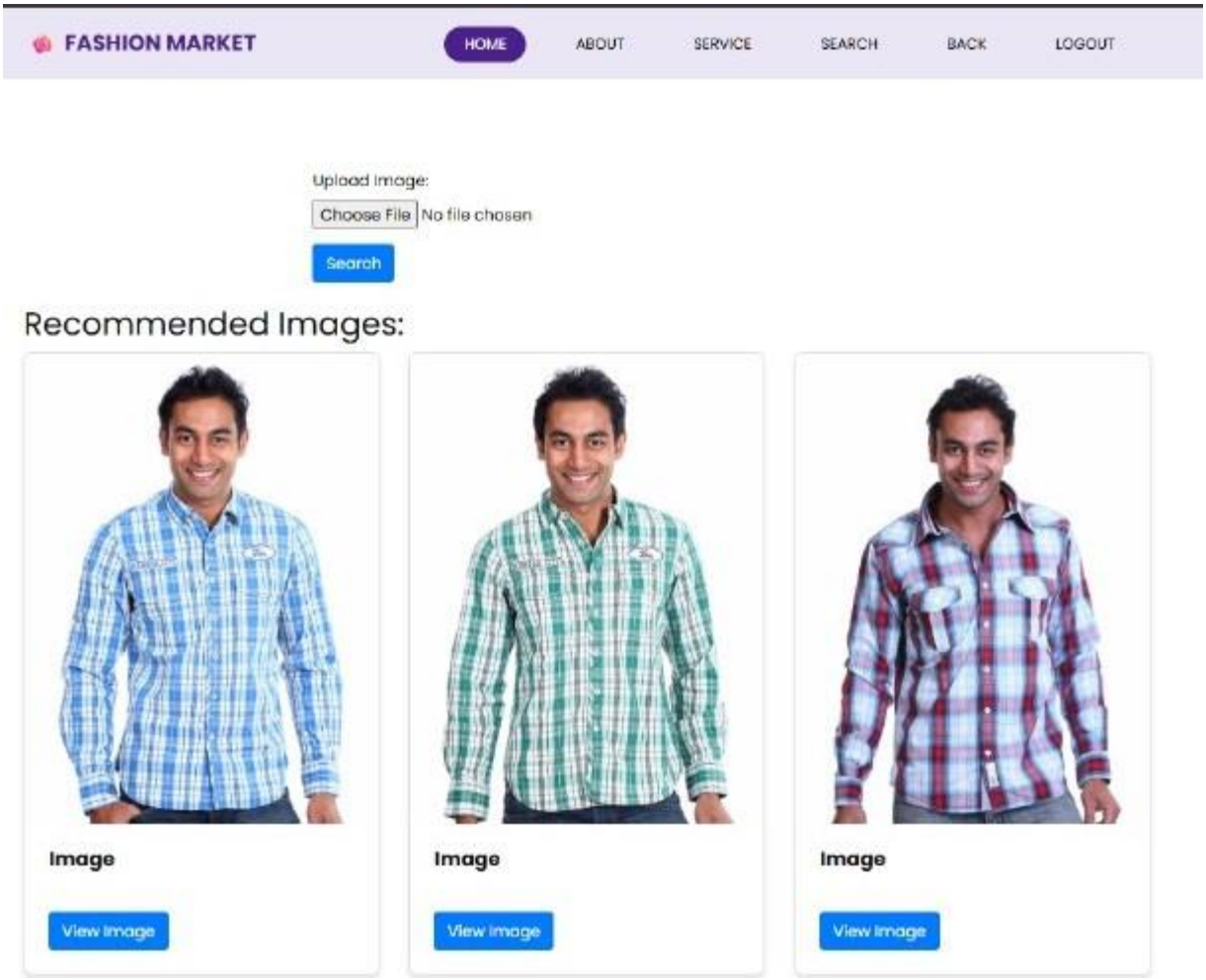


Fig: 7.2.6 Result Page

Based on the user's Instagram profile picture, his clothing is processed through the dataset using the algorithms like cnn and densenet and a set of similar clothes is recommended to the user. This is the result page based on the users input

CHAPTER 8
SOFTWARE TESTING

8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3 TYPES OF TESTS

8.3.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3.2 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

8.3.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 Performance Testing

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance Testing for Data Synchronization

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

8.2.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 9
CONCLUSION AND FUTURE ENHANCEMENT

9.1 CONCLUSION

Valued at over 3 trillion dollars, the global fashion industry contributes to a healthy 2% of the global Gross Domestic Product (GDP). For this reason, fashion companies are increasingly trying to invest in the world of artificial intelligence to be able to satisfy the customer 100%. In particular, social media have long since changed the way of perceiving the world of fashion by the costumers: in this context social networks are fundamental communication tools, in particular Facebook and Instagram. Above all, the Instagram social network has become of fundamental importance for companies as the influencer sponsoring products is paid by companies to influence consumer preferences.

For this reason, this review aims to summarize the datasets that have been collected and the methods that have been used in deep learning in the fashion sector, and in particular in social networks. Methods and techniques for each kind of fashion task have been analysed, the main paths have been summarised, and their contributions have been highlighted. This review offers rich information and improves the understanding of the research issues related to the use of AI with social media fashion data. Furthermore, it is informative on how and if DL techniques and methods could help the development of applications in various fields.

9.2 FUTURE ENHANCEMENT

Future research directions include the improvement of the algorithms to use other comprehensive features, thereby achieving better performance

REFERENCES

[1] A. Adewumi, A. Taiwo, S. Misra, R. Maskeliunas, R. Damasevicius, R. Ahuja, and F. Ayeni, ‘‘A unified framework for outfit design and advice,’’ in Data Management, Analytics and Innovation. Springer, 2020.

[2] T. H. Nobile, A. Noris, N. Kalbaska, and L. Cantoni, ‘‘A review of digital fashion research: Before and beyond communication and marketing,’’ Int. J. Fashion Des., Technol. Educ., pp. 1–9, May 2021.

[3] M. Paolanti and E. Frontoni, ‘‘Multidisciplinary pattern recognition applications: A review,’’ Comput. Sci. Rev., vol. 37, Aug. 2020.

[4] X. Gu, F. Gao, M. Tan, and P. Peng, ‘‘Fashion analysis and understanding with artificial intelligence,’’ Inf. Process. Manage., vol. 57, no. 5, Sep. 2020.

[5] C. Giri, S. Jain, X. Zeng, and P. Bruniaux, ‘‘A detailed review of artificial intelligence applied in the fashion and apparel industry,’’ IEEE Access, vol. 7, pp. 95376–95396, 2019.

[6] L. Q. Lomas, A. G. Elordi, A. A.Escondrillas, and D. L. De Ipina Gonzalez De Artaza, ‘‘A systematic literature review of artificial intelligence in fashion retail B2C,’’ in Proc. 6th Int. Conf. Smart Sustain. Technol. (SpliTech), Sep. 2021.

[7] S. Chakraborty, M. Hoque, N. Rahman Jeem, M. C. Biswas, D. Bardhan, and E. Lobaton, ‘‘Fashion recommendation systems, models and methods: A review,’’ Informatics, vol. 8, no. 3, p. 49, 2021.

[8] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, ‘‘Parsing clothing in fashion photographs,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012.

[9] J. Dong, Q. Chen, W. Xia, Z. Huang, and S. Yan, ‘‘A deformable mixture parsing model with parselets,’’ in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013.

[10] K. Yamaguchi, M. H. Kiapour, and T. L. Berg, ‘‘Paper doll parsing: Retrieving similar styles to parse clothing items,’’ in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013.

[11] W. Yang, P. Luo, and L. Lin, ‘‘Clothing co-parsing by joint image segmentation and labeling,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014.

[12] S. Liu, J. Feng, C. Domokos, H. Xu, J. Huang, Z. Hu, and S. Yan, ‘‘Fashion parsing with weak color-category labels,’’ IEEE Trans. Multimedia, vol. 16, no. 1, pp. 253–265, Jan. 2013.

[13] B. Loni, L. Y. Cheung, M. Riegler, A. Bozzon, L. Gottlieb, and M. Larson, ‘‘Fashion 10000: An enriched social image dataset for fashion and clothing,’’ in Proc. 5th ACM Multimedia Syst. Conf. (MMSys), 2014.

[14] J. Huang, W. Xia, and S. Yan, ‘‘Deep search with attribute-aware deep network,’’ in Proc. 22nd ACM Int. Conf. Multimedia, Nov. 2014.

[15] A. Yu and K. Grauman, ‘‘Fine-grained visual comparisons with local learning,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014.

[16] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan, ‘‘Deep human parsing with active template regression,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 12, pp. 2402–2414, Dec. 2015.

[17] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, and S. Yan, ‘‘Human parsing with contextualized convolutional neural network,’’ in Proc. IEEE Int. Conf. Comput. Vis., Jun.2015.

[18] J. Huang, R. Feris, Q. Chen, and S. Yan, ‘‘Cross-domain image retrieval with a dual attribute-aware ranking network,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015.

[19] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, ‘‘Where to buy it: Matching street clothing photos in online shops,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015.

[20] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, ‘‘DeepFashion: Powering robust clothes recognition and retrieval with rich annotations,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016.

[21] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang, ‘‘Fashion landmark detection in the wild,’’ in Proc. Eur. Conf. Comput. Vis. Springer, 2016.

[22] K.-H. Liu, T.-Y. Chen, and C.-S. Chen, ‘‘MVC: A dataset for view-invariant clothing retrieval and attribute prediction,’’ in Proc. ACM Int. Conf. Multimedia Retr., Jun. 2016.

[23] Z. Li, Y. Li, W. Tian, Y. Pang, and Y. Liu, ‘‘Cross-scenario clothing retrieval and fine-grained style recognition,’’ in Proc. 23rd Int. Conf. Pattern Recognit. (ICPR), Dec. 2016.

[24] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, ‘‘Pixel-level domain transfer,’’ in Proc. Eur. Conf.Comput. Vis. Springer, 2016.

[25] S. Yan, Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, ‘‘Unconstrained fashion landmark detection via hierarchical recurrent transformer networks,’’ in Proc. 25th ACM Int. Conf. Multimedia, Oct. 2017.

[26] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin, ‘‘Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR), Jul. 2017.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, ‘‘Microsoft coco: Common objects in context,’’ in Proc. Eur. Conf. Comput. Vis. Springer, 2014.

[28] F. Xia, P. Wang, X. Chen, and A. L. Yuille, ‘‘Joint multi-person pose estimation and semantic part segmentation,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017.

[29] J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, T. Sim, S. Yan, and J. Feng, ‘‘Multiple-human parsing in the wild,’’ 2017.

[30] Z.-Q. Cheng, X. Wu, Y. Liu, and X.-S. Hua, ‘‘Video2Shop: Exact matching clothes in videos to online shopping images,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017.

[31] N. Garcia and G. Vogiatzis, ‘‘Dress like a star: Retrieving fashion products from videos,’’ in Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW), Oct. 2017.

[32] H. Xiao, K. Rasul, and R. Vollgraf, ‘‘Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,’’ 2017.

[33] M. Takagi, E. Simo-Serra, S. Iizuka, and H. Ishikawa, ‘‘What makes a style: Experimental analysis of fashion prediction,’’ in Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW), Oct. 2017.

[34] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, ‘‘Learning fashion compatibility with bidirectional LSTMs,’’ in Proc. 25th ACM Int. Conf. Multimedia, Oct. 2017.

[35] N. Jetchev and U. Bergmann, ‘‘The conditional analogy GAN: Swapping fashion articles on people images,’’ in Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW), Oct. 2017.

[36] J. Zhao, J. Li, Y. Cheng, T. Sim, S. Yan, and J. Feng, ‘‘Understanding humans in crowded scenes: Deep nested adversarial learning and a new benchmark for multi-human parsing,’’ in Proc. 26th ACM Int. Conf. Multimedia, Oct. 2018.

[37] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin, ‘‘Instance-level human parsing via part grouping network,’’ in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018.
[38] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu, ‘‘ModaNet: A large-scale street fashion dataset with polygon annotations,’’ in Proc. 26th ACM Int. Conf. Multimedia, Oct. 2018.
[39] L. Liao, X. He, B. Zhao, C.-W. Ngo, and T.-S. Chua, ‘‘Interpretable multimodal retrieval for fashion products,’’ in Proc. 26th ACM Int. Conf. Multimedia, Oct. 2018.
[40] N. Rostamzadeh, S. Hosseini, T. Boquet, W. Stokowiec, Y. Zhang, C. Jauvin, and C. Pal, ‘‘Fashion-gen: The generative fashion dataset and challenge,’’ 2018.
[41] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, ‘‘VITON: An image-based virtual try-on network,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018.
[42] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, ‘‘DeepFashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019.
[43] Z. Kuang, Y. Gao, G. Li, P. Luo, Y. Chen, L. Lin, and W. Zhang, ‘‘Fashion retrieval via graph reasoning networks on a similarity pyramid,’’ in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019.
[44] X. Wang, B. Wu, and Y. Zhong, ‘‘Outfit compatibility prediction and diagnosis with multi-layered comparison network,’’ in Proc. 27th ACM Int. Conf. Multimedia, Oct. 2019.
[45] S. Guo, W. Huang, X. Zhang, P. Srikhanta, Y. Cui, Y. Li, H. Adam, M. R. Scott, and S. Belongie, ‘‘The iMaterialist fashion attribute dataset,’’ in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW), Oct. 2019.
[46] K.-H. Liu, F. Wang, and T.-J. Liu, ‘‘A clothing recommendation dataset for online shopping,’’ in Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-TW), May 2019.
[47] X. Zou, X. Kong, W. Wong, C. Wang, Y. Liu, and Y. Cao, ‘‘FashionAI: A hierarchical dataset for fashion understanding,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2019.
[48] H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris, ‘‘Fashion IQ: A new dataset towards retrieving images by natural language feedback,’’ 2019.
[49] N. Zheng, X. Song, Z. Chen, L. Hu, D. Cao, and L. Nie, ‘‘Virtually trying on new clothing with arbitrary poses,’’ in Proc. 27th ACM Int. Conf. Multimedia, Oct. 2019.

[50] C.-W. Hsieh, C.-Y. Chen, C.-L. Chou, H.-H. Shuai, J. Liu, and W.-H. Cheng, ‘‘FashionOn: Semantic-guided image-based virtual try-on with detailed human and clothing information,’’ in Proc. 27th ACM Int. Conf. Multimedia, Oct. 2019.

[51] H. Dong, X. Liang, X. Shen, B. Wu, B.-C. Chen, and J. Yin, ‘‘FW-GAN: Flow-navigated warping GAN for video virtual try-on,’’ in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019.

[52] Y. Ma, X. Yang, L. Liao, Y. Cao, and T.-S. Chua, ‘‘Who, where, and what to wear?: Extracting fashion knowledge from social media,’’ in Proc. 27th ACM Int. Conf. Multimedia, Oct. 2019.

[53] X. Yang, H. Zhang, D. Jin, Y. Liu, C.-H. Wu, J. Tan, D. Xie, J. Wang, and X. Wang, ‘‘Fashion captioning: Towards generating accurate descriptions with semantic rewards,’’ 2020.

[54] T. Kishore Babu; Raja Kiran Kolati; Pathipati Chandrasekhar; Nimmagadda MuraliKrishna; Sriharsha Vikruthi; B.Rajeswari,”Computer-Assisted Leukemia Detection and Classification using Machine Learning”,2024 International Conference on Expert Clouds and Applications (ICOECA),2024

https://www.geeksforgeeks.org/download-instagram-posts-using-python-selenium-module/