FASHION RECOMMENTED SYSTEM USING PYTHON

CHAPTER 1

INTRODUCTION

The world is transitioning to an "E-world" where the majority of things are digitized and accessible with a mouse click thanks to the emergence of developing technologies and the Internet's explosive expansion. The role of e-commerce in transforming consumer purchasing behavior is gradually growing. More and more things are being purchased by customers on the internet. When a client wants to purchase a product online, they go to an online store and search for the products they are interested in. Online E-commerce applications are widely available at the moment. Every programmed has a common flaw: there is no customer service. We have developed a method for product recommendations to address this issue. The topic of recommendation systems is rapidly developing in the realm of online services and e-commerce applications.

It takes a lot of time to search through numerous things while purchasing online. A recommendation system helps expedite the process of locating a large range of goods that clients are interested in. As it is simple and dependable for a consumer to purchase online and locate the ideal solutions for them without any hassles, the use of this effective suggestion system is growing day by day.

Systems that provide recommendations are advantageous to both consumers and service providers. Previous studies have shown that using recommendation engines may improve customer decision-making, cut down on search time, and help consumers locate the best deals.

A product recommendation system is a piece of software made to come up with ideas for things a certain user would want to buy or engage with. We concentrate on fashion items and design a system that just needs one input

image to produce a ranked list of similar-style recommendations. Product recommendation will evaluate the current things more thoroughly.

The construction of an e-commerce application for the selling of various product kinds grouped under comparable features is the subject of this article. Utilizing Transfer Learning and cosine similarity, the Deep Learning approach provides the most comparable products in relation to the user's product selections. The system's features include ways to search for related items and product data. Sending the consumer an email with information about their purchases and providing an anticipated delivery date for each good are other implementations.

Figure 1.1 fashion recommended system

Customers are frustrated by the online store's lack of recommendation tools and their time-consuming searches. recognizable attire that matched the frontal aspect outfit pictures. has suggested the garments to the buyers based on their prior shopping behavior.

WHAT IS E-COMMERCE

E-commerce is the process of selling goods and services over the internet. Customers come to the website or online marketplace and purchase products using electronic payments. Upon receiving the money, the merchant ships the goods or provides the service.

Figure 1.2 Architecture of E-commerce

Electronic commerce has been around since the early 1990s when Amazon just sold books, but today, it's a multibillion-dollar industry – and it has gotten even bigger during the pandemic. According to Digital Commerce 360's analysis of U.S. Department of Commerce data, e-commerce spending hit $347.26 billion in the first half of 2020, up 30% year over year. For comparison, e-commerce sales only increased 12.7% during the first six months of 2019.

WORKING OF E-COMMERCE WORK

E-commerce works on the same principles as a physical store. Customers come into your e-commerce store, browse products and make a purchase. The big difference is they don't have to get off their couch to do so, and your customer base isn't limited to a specific geographic area or region.

1. **Accept the Order:** The Customer Places an Order on Your Website or E-Commerce Platform. You'll Be Alerted That an Order Was Placed.
2. **Process the Order:** Next, The Payment Is Processed, The Sale Is Logged, And the Order Is Marked Complete. Payment Transactions Are Usually Processed Through What Is Known as A Payment Gateway; Think of It as The Online Equivalent of Your Cash Register
3. **Ship the Order:** The Last Step in The E-Commerce Process Is Shipment. You Have to Ensure Prompt Delivery If You Want Repeat Customers.

CHATBOTS

Over the last few years, use of mobile messaging apps such as Facebook Messenger and WhatsApp has exploded in popularity, and thanks to ecommerce personalization, many large retailers have integrated AI. Now, these two trends have converged in the Chatbot – AI-led automated messenger services that allow customers to engage with a brand, delivering a new way of shopping and communicating. The best Chatbots detect exactly what the customer is looking for, make suggestions about products, answer customer service queries and place orders, even detecting idioms in language used in order to communicate in an appropriate manner.

BENEFITS OF CHATBOT

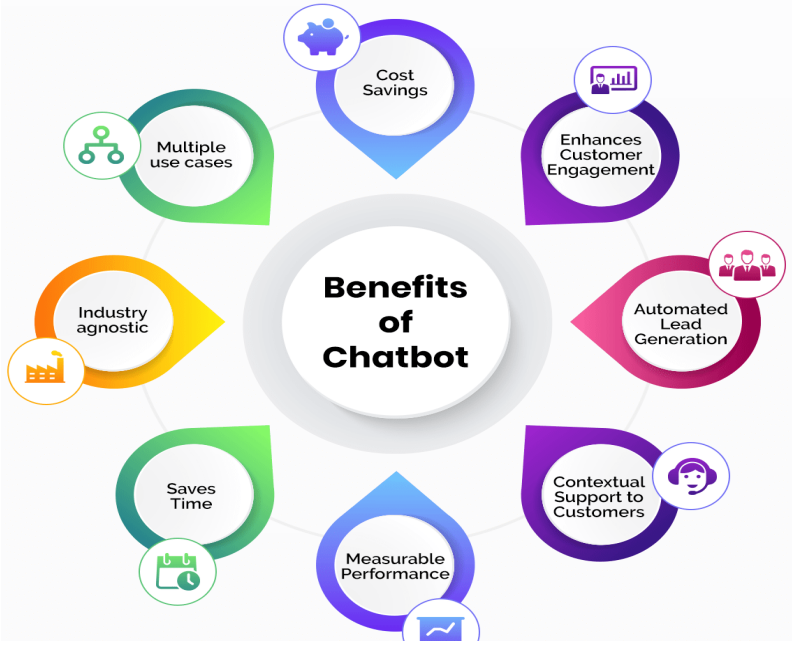


Figure 1.3 Benefits of chatbot

- Enhances customer engagement and sales
- Better lead generation, qualification & nurturing
- Bots save a great deal of time
- Cost savings
- Offer website visitors contextual, ai-driven support
- Better analysis of customer data.

CHAPTER 2

ABSTRACT

Now a days, fast fashion has seen considerably stronger in the textile and garment industries. As the level of daily existence increased, people's focus progressively shifted to fashion, which is seen to be a common form of aesthetic expression. Unavoidably, humans are drawn to things that are more appealing on the surface. This human propensity has contributed to the growth of the fashion industry over time. But the excess of clothing alternatives on e-commerce platforms has made it harder for buyers to choose the right outfit. So, in this work, we suggested a method for creating personalized fashion recommendations for the user based on input. In contrast to the conventional systems, which rely on a user's past purchases and history, this project aims to generate recommendations using an image of a product submitted by the user. This is because many times, when people see something, they like, they tend to look for products that are similar to that. The Deep Fashion dataset's photos are processed using a neural network in Python, and the final suggestions are produced using a nearest neighbours backed recommender. The recommended fashion system is very useful of Convert Shoppers to Customers, Increase Average Order Value, Increase Number of Items per Order, Control Merchandising and Inventory Rules, Reduce Workload and Overhead.

CHAPTER 3

LITERATURE SURVEY

1. **M Sridevi et al (2020):** Thus, in this paper, we proposed a personalized Fashion Recommender system that generates recommendations for the user based on an input given. Unlike the conventional systems that rely on user's previous purchases and history, this project aims at using an image of a product given as input by the user to generate recommendations since many-a-time people see something that they are interested in and tend to look for products that are similar to that. We use neural networks to process the images from Deep Fashion dataset and a nearest neighbour backed recommender to generate the final recommendations. we propose a novel personalized Fashion Recommender system on the basis of user preferences. More specifically, we focus on the features of fashion and develop a framework that accepts a single image as an input and returns a top -5 ranked list of similar clothing recommendations.
2. **Manish Pathak et al (2021):** Recommendation engines are integral to the modern e-commerce experience, both for the seller and the end user. Accurate recommendations lead to higher revenue and better user experience. In this paper, we are presenting our solution to ECML PKDD Far fetch Fashion Recommendation Challenge. The goal of this challenge is to maximize the chances of a click when the users are presented with set of fashion items. We have approached this problem as a binary classification problem. Our winning solution utilizes Cat boost as the classifier and Bayesian Optimization for hyper parameter tuning. Our baseline model achieved MRR of 0.5153 on the validation set. Bayesian optimization of hyper parameters improved the MRR to 0.5240 on the validation set. Our final submission on the test set achieved a MRR of 0.5257.

3. **Muhammad KHALID et al (2021):** This research designs and implements two-stage deep learning-based model that recommends a clothing fashion style. This model can use deep learning approach to extract various attributes from images with clothes to learn the user's clothing style and preferences. These attributes are provided to the correspondence model to retrieve the contiguous related images for recommendation. Based on data-driven, this thesis uses convolutional neural network as a visual extractor of image objects. This experimental model shows and achieves better results than the ones of the previous schemes.
4. **Batuhan AS¸ IROGLU et al (2019):** In this study, we develop a cloth recommendation system with using only single photo of user with scalable embedded system. This study led to important results and give new opportunities for clothing companies and advertisements. In this study, we show that how our system recommends a cloth option without user's previous shopping act data with embedded system and machine learning. In order to recommend a cloth, we develop two inception based convolutional neural networks as prediction part and one feed forward neural network as recommender. In this study, we reach to 98% accuracy on colour prediction, 86% accuracy on gender and cloth's pattern predictions and 75% accuracy on clothing recommendation.
5. **Narges Yarahmadi Gharaei et al (2021):** In this paper, we propose a content-based clothing recommender system using deep neural network. In content-based systems, product features are required for prediction of unobserved items ratings. In our proposed system by using a deep neural network, the cloth category is obtained and the need to manually extract the product features is eliminated by producing the required features with a large and useful volume. The advantage of this system is that it uses the same network to specify gender as a feature in making suggestions then shows the results to the user. Different machine learning algorithms are tested and

analysed with and without considering demographic information such as gender. The experimental results show that the loss of our proposed system is lower than the other related systems and solves the cold start problem for new items. Our proposed system also recommends novel, relevant and unexpected items.

6. **Illa Pavan Kumar et al (2019):** This paper aims to simulate this recommendation system on real world data set taken from the marketing giant, Amazon's Product Advertising API, in a policy compliant manner by following the procedure in three steps: Analysing the data to select the pivot for the recommendation system, Data pre-processing to remove invalid sections and to implement and find appropriate choices among the techniques like Bag of Words (Bow) and TF-IDF for better recommendations. As a result, there is a necessity and a great purpose to stumble on a set of objective indicators, rather to concentrate on subjective opinions, for evaluating role of recommendation technology in apparel industry. Thus, recommendation system plays a vital role in e-commerce applications like movies, shopping, tourism, TV, taxi.
7. **Samit Chakraborty et al (2021):** With the technological advancements, this branch of artificial intelligence exhibits a tremendous amount of potential in image processing, parsing, classification, and segmentation. Despite its huge potential, the number of academic articles on this topic is limited. The available studies do not provide a rigorous review of fashion recommendation systems and the corresponding filtering techniques. To the best of the authors' knowledge, this is the first scholarly article to review the state-of-the-art fashion recommendation systems and the corresponding filtering techniques. In addition, this review also explores various potential models that could be implemented to develop fashion recommendation systems in the future. This paper will help researchers, academics, and practitioners who are interested in machine learning, computer vision, and

fashion retailing to understand the characteristics of the different fashion recommendation system.

8. **Reshma B Nair et al (2018):** It provides information in all fields such as education, sports, fashion, etc and also updates us with all the latest news that is happening around the world. Some of the popular social media sites are Facebook, Twitter, Instagram and it has become a platform for people to easily communicate and share information. Nowadays business people use social media as their means for communication. Fashion industry is one of the businesses which frequently changes and the social media is considered to be as the cheapest and easiest means to communicate. Hence an application is created which obtains the photos of the user from their Facebook account and suggests the similar kind of dresses based on their collection of dresses.
9. **Akshit Tayade et al (2021):** By the use of Transfer Learning to elicit the rich information from the product images, and the use of cosine similarity approach, the user is provided with eclectic recommended products depending on their choices. This was implemented on a web e-commerce application, where users have a wide range to choose their clothing apparel from our database. Only those images are recommended to users which have 80% or above similarity with the product chosen by the user, which helps in solving personalized recommendations. The features of the system include product details and similar item search methods. Additional implementation includes sending an email to the user about their purchased items and giving an estimated date of delivery for individual products.
10. **Gursimran Kaur, et al (2021):** There is an introduction to the three types of Recommendation Systems that are present: Content based, Collaborative Filtering and Hybrid Models, and a discussion on their pros and cons. Then onto discussing the challenges faced by Recommendation approaches followed by specifically the ones by Fashion Recommendation Systems. The

need for presenting Outfit recommendation models and the importance of their accuracy is presented. Finally, a comprehensive survey of 4 types of Fashion Recommendation Systems: 1.) Collaborative Filtering 2.) Content based 3.) Hybrid 4.) Ontology based. A presentation of these with examples for representative algorithms of each category, and analysis of their predictive performance and their ability to address the challenges is carried out.

11. **Seema Wazarkar et al (2022):** These recommendation models lack personalization based on the user's body demographics. Since fashion is a way, one chooses to express themselves, it is important that each piece is carefully selected to suit the buyer. In this paper, an improved recommendation system is developed using a deep learning model for customers with different body shapes/types. It helps users to select clothing items based on their body shape. Proposed system is evaluated with respect to multiple deep learning models as well as traditional machine learning approaches. Exception model out performed by achieving 94% accuracy and a loss of 0.02%.

CHAPTER 4

EXISTING SYSTEM

In this paper, we propose a model that uses Convolutional Neural Network and Nearest neighbour backed recommender. Initially, the neural networks are trained and then an inventory is selected for generating recommendations and a database is created for the items in inventory. The nearest neighbour's algorithm is used to find the most relevant products based on the input image and recommendations are generated.

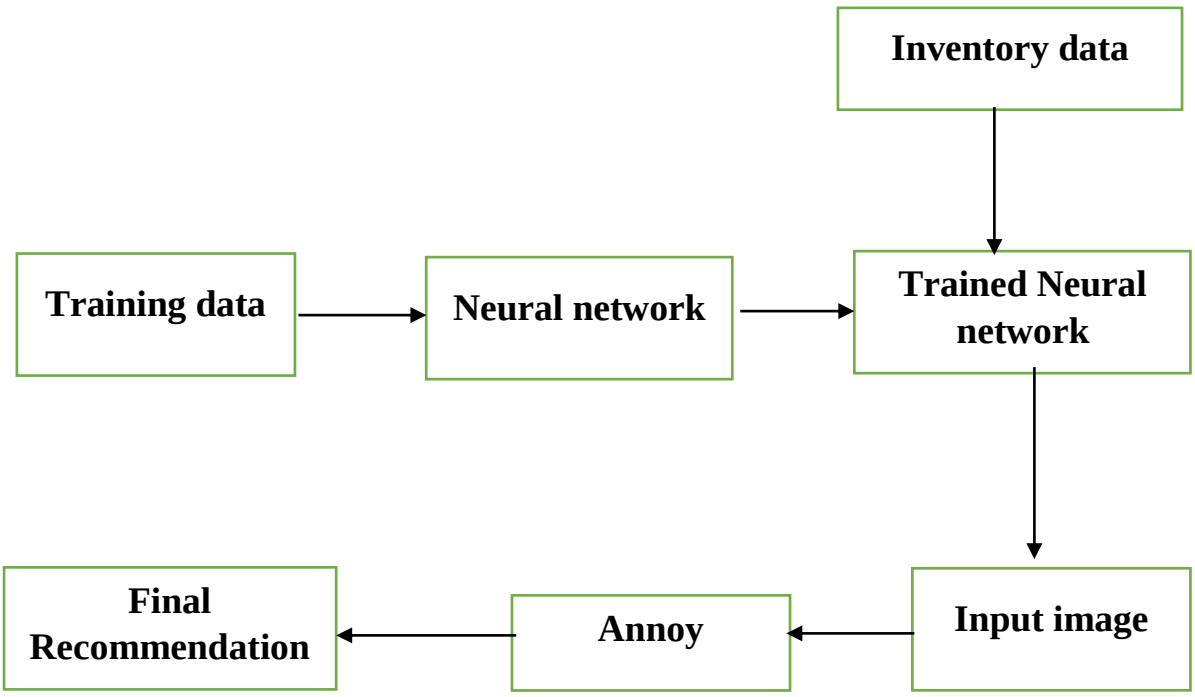


Figure 4.1 Block diagram of existing system

Once the data is pre-processed, the neural networks are trained using fastai, utilizing transfer learning from ResNet50. More additional layers are added in the last layers that replaces the architecture and weights from ResNet50 in order to fine tune the network model to serve the current issue.

The framework for fashion recommendation that is driven by data, visually related and simple effective recommendation systems for generating fashion product images. This approach uses a two-stage phase and extracts the features of the image using CNN classifier ie., for instance allowing the customers to upload any random fashion image from any Ecommerce website and later generating similar images to the uploaded image based on the features and texture of the input image. It is imperative that such research goes forward to facilitate greater recommendation accuracy and improve the overall experience of fashion exploration for direct and indirect consumers alike.

DISADAVANTAGES OF EXISTING SYSTEM:

- They fail to encode the position and orientation of objects.
- A lot of training data is needed for the CNN to be effective.
- CNNs tend to be much slower because of operations like maxpool.

CHAPTER 5

PROPOSED SYSTEM

We utilized the special Kaggle dataset, which includes fashion apparel including shoes, heels, rings, and garments, to obtain eccentric clothing apparel. Although there were many different types of fashion items in this dataset, we only used clothing for men and women. A text file with fields for the brand name, product name, product type, label, and picture file are included with the dataset. We remove any item rows that were unnecessary by using the column: product type. Our database now contained just the most important and cleanly taken pictures of clothes. We also used the Deep Fashion Database's Large-scale clothing dataset, which was retrieved. Using the Deep Fashion database, four benchmarks are created for attribute prediction, consumer-to-shop clothes retrieval, in-shop clothes retrieval, and landmark detection.

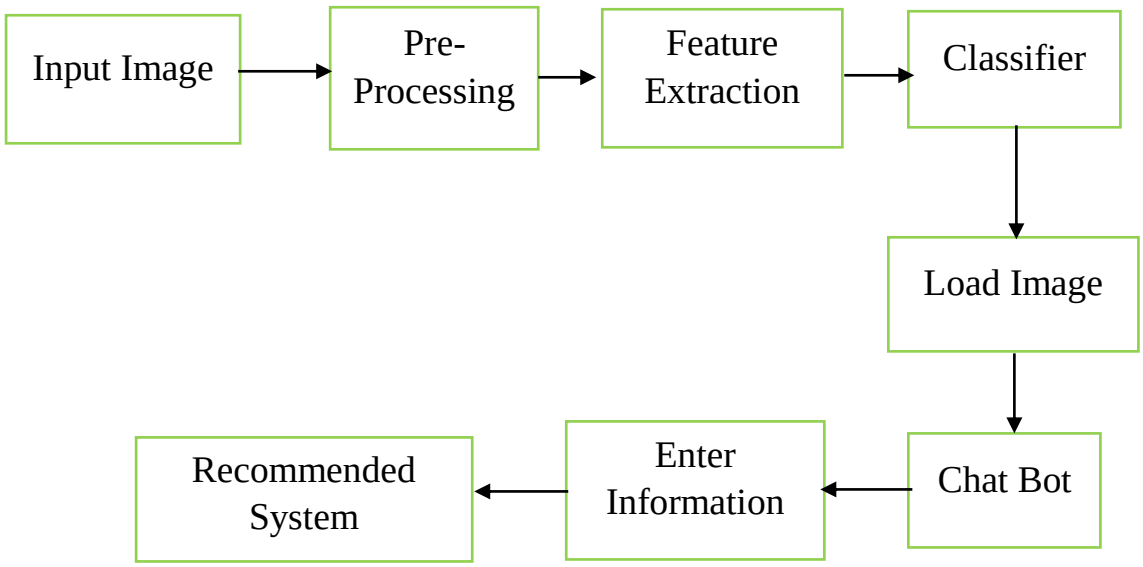


Figure 5.1 Block Diagram of Proposed System

For the following computer vision tasks, such as Clothes Detection, Clothes Recognition, and Image Retrieval, the data and annotations of these benchmarks are used as the training and test sets. This dataset includes a variety of fashion photos, including well-posed store shots with 14 distinct female and 9 different male outfits. Separating male and female objects, we merged more than 1000 photos.

CHAPTER 6

SYSTEM REQUIREMENTS

Hardware Requirements

- CPU type : Intel Pentium 4
- Clock speed : 3.0 GHz
- Ram size : 8 GB
- Hard disk capacity: 500 GB
- Monitor type : 15 Inch colour monitor

Software Requirement

- Operating System : Windows 10
- Language : Python
- Tool : Anaconda

Software Description

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. Python is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain.

ADVANTAGES OF LEARNING 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.

Characteristics of Python

Following are important characteristics of Python Programming

- 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.

Python Features

Python provides lots of features that are listed below.

1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high-level programming language.

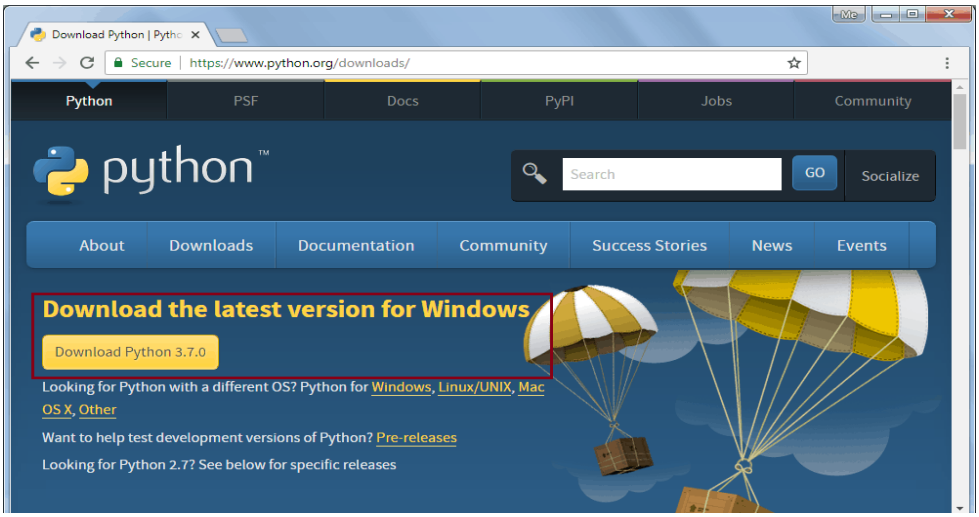


Figure 6.1 Install of python

2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

5) Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence.

6) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

7) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

8) GUI Programming Support

Graphical user interfaces can be developed using Python.

9) Integrated

It can be easily integrated with languages like C, C++, and JAVA etc.

Python Applications

Python is known for its general-purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.

Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautiful Soup, Feed parser etc. It also provides Frameworks such as Django, Pyramid, Flask etc. to design and develop web-based applications.

Desktop GUI Applications

Python provides Tk GUI library to develop user interface in python-based application. Some other useful toolkits wx Widgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multi touch applications.

Software Development

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

Scientific and Numeric

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.

Business Applications

Python is used to build Business applications like ERP and e-commerce systems.

Console Based Application

We can use Python to develop console-based applications. For example: IPython.

Audio or Video based Applications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

3D CAD Applications

To create CAD application Fandango is a real application which provides full features of CAD.

Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

Applications for Images

Using Python several applications can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc.

CHAPTER 7

SYSTEM TESTING

TESTING PRINCIPLES

Before applying method to design effective test cases, a software engineer must understand the basic principles that guide software testing. Davis (DAV95) suggests a set of testing principles which have been adapted for use in this book.

- All tests should be traceable to customer requirements.
- Test should be planned long before testing begins.
- Test pare to principal applets to software testing. Testing should begin “in the small” and progress towards testing “in the page”.
- Exhaustive testing is not possible.

Unit Testing

Unit testing focuses on verification errors on the smallest unit of software design-the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

The module interface is tested to ensure that the information properly flows into and out of the program unit under test. Boundaries conditions are tested to ensure that the module operates properly at the boundaries established to limit of restrict processing.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design.

White Box Testing

White box testing is some time is called glass box testing, is a test case design that uses a control structure of the procedural design to drive the test cases. Using white-box testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions are on the true and false sides
- Exercise all logical decisions are on the true and false sides
- Execute all loops at their boundaries and within their operational bounds
- Exercise internal data structure to assure the validity

Acceptance Testing

Finally, when the software is completely built, a series of acceptance tests are conducted to enable the client to validate all requirements. The user conducts these tests rather than the system developer, which can range from informal test drive to a planned and systematically executed series of tests. These acceptance tests are conducted over a period of weeks or months, there by uncovering cumulative errors that might degrade the system order time. In this process alpha testing and beta testing are used to uncover the errors that only the end user seems able to find.

Alpha Testing

The customer conducts the alpha test at the developer's site. The client notes the errors and usage problems and gives report to the developer. Alpha tests are conducted in a control environment.

Beta Testing

The beta testing is conducted at one or more customer's sites by the end users of the software. Unlike the alpha testing, the developer is not present. Therefore, a beta test is a "live" application of the software in the environment that cannot be

developed by the developer. The customer records all the problems encountered during the beta testing and reports these to the developers at regular intervals.

Black Box Testing

Black box testing focuses on the functional requirements of the software. That is black box testing enables the software engineer to drive a set of input conditions that will fully exercise the requirements for a program.

Black box testing is not an alternative for white box testing techniques. Rather, it is a complementary approach that is likely to uncover different class of errors. Black box testing attempts to find errors in the following categories:

- Interface errors.
- Performances in data structures or external database access.
- Performance errors.
- Initialization and termination errors.
- Incorrect or missing functions.

All the above-mentioned errors were checked in the process of black box testing and the bugs Found Were Fixed.

Test Cases

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to your customer. Your goal is to design a series of test cases that have a high likelihood of finding errors. To do so we have techniques provide systematic guidance for designing tests that: the internal logic of software components, and the input and output domains of the program to uncover errors in program function, behaviour, and performance. Resource presented in this section address the following topic categories. Software Testing is the process of confirming the functionality and correctness of software by running it. Software testing is usually performed for one of two reasons:

- Defect detection
- Reliability estimation.

The problem of applying software testing to defect detection is that software can only suggest the presence of flaws, not their absence (unless the testing is exhaustive). The problem of applying software testing to reliability estimation is that the input distribution used for selecting test cases may be flawed. In both of these cases, the mechanism used to determine whether program output is correct is often impossible to develop. Obviously, the benefit of the entire software testing process is highly dependent on many different pieces. If any of these parts is faulty, the entire process is compromised.

Software is now unique unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible.

The key to software testing is trying to find the myriad of failure modes – something that requires exhaustively testing the code on all possible inputs. For most programs, this is computationally infeasible. It is commonplace to attempt to test as many of the syntactic features of the code as possible (within some set of resource constraints) are called white box software testing technique. Techniques that do not consider the code's structure when test cases are selected are called black box technique.

Functional testing is a testing process that is black box in nature. It is aimed at examine the overall functionality of the product. It usually includes testing of all the interfaces and should therefore involve the clients in the process. Final stage of the testing process should be System Testing. This type of test involves examination of the whole computer system, all the software components, all the

hard ware components and any interfaces. The whole computer-based system is checked not only for validity but also to meet the objectives.

System Implementation

Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system or it may be major modification to the system currently put into use. This system "Access Point Selection for Improving the Voice Quality and Overall Throughput in Wireless LANs" is a new system. Implementation as a whole involves all those tasks that we do for successfully replacing the existing or introduce new software to satisfy the requirement. The test case has performed in all aspect and the system has given correct result in all the cases.

The System implementation phase consists of the following steps:

- Testing the developed software with sample data.
- Correction of any errors if identified.
- Creating the files of the system with actual data.
- Making necessary changes to the system to find out errors.
- Training of user personnel.

The system has been tested with sample data, changes are made to the user requirements and run in parallel with the existing system to find out the discrepancies. The user has also been appraised how to run the system during the training period.

This phase is primarily concerned with user training, site preparation and file conversions. During the final testing, user acceptance is tested, followed by user training. Depending in the nature of the extensive user training may be required.

After development and testing has been completed, implementation of the information system can begin. During system implementation, the project team should be brought back to full strength. During software development stage, project

teams end to play passive role as the technical steps of program development and testing evolve. However, broad organizational representation, accomplished through the project team, is required to complete the system development cycle.NET Framework has offer very efficient yet simple implementation techniques for development of the project.

Implementation plan

Implementation is the stage, which is crucial in the life cycle of the new system designed. Implementation means converting a new or revised system design into an operational one. This is the stage of the project where the theoretical design is turned into a working system. In this project "Access Point Selection" implementation includes all those activities that take place to convert from the old system to the new one. The important phase of implementation plan is change over.

The implementation phase's construction, installation and operations lie on the new system. The most crucial and very important stage in achieving a new successful system and in giving confidence on the new system for the user that it will work efficiently and effectively.

There are several activities involved while implementing a project:

- Careful planning
- Investigation current system and its constraints on implementation
- Design of methods to achieve the change over
- Training of the staff in the changeover procedure and evaluation of change over method

The implementation is the final stage and it is an important phase. It involves the individual programming system testing, user training and the operational running of developed proposed system that constitutes the application subsystems. On major task of preparing for implementation is education of users, which would really have taken place much earlier in the project when we're being involved in

the investigation and design work. The implementation phase of software development is concerned with translating design specifications into source code. The user tests the developed system and changes are made according to their needs.

Change over

The implementation is to be done step by step since testing with dummy data will not always reveal the faults. The system will be subjected to the employees to work. If such error or failure is found, the system can be corrected before it is implemented in full stretch. The trail should be done as long as the system is made sure to function without any failure or errors. Precautions should be taken so that any error if occurred should not totally make the process to a halt. Such a care should be taken. The system can be fully established if it does not create any error during the testing period.

Education and user training

Well-designed and technically elegant systems can succeed or fail because of the way they are operated and used. Therefore, the quality of the training received by the personnel involved with the systems help or hinder, and may even prevent, the successful completion of the system.

An analysis of user training focuses on user capabilities and the nature of the system being installed. Those users are verifying type and nature. Some of them may not have any knowledge about the computers and the others may be very intelligent. The requirements of the system also range from simple to complex tasks. So the training has to be generated to the specific user based on his/her capabilities and system's complexity. User training must instruct individuals in trouble shooting the system, determining whether a problem that arises is caused by hardware or software. A good or perfect documentation which instructs the user on how to start the system and the various functions and meanings of various codes must be prepared and that will help the user to understand the system in a better

manner. Through the training demonstration with personnel contact also, the user can be trained. This training demonstration will help the users to understand the system in many ways.

By this the user receives encouragement and attention. Another rapid way of training the user is by resident experts. Several user training aids are provided such as user manual.

CHAPTER 8

FEASIBILITY STUDY

TECHNOLOGY AND SYSTEM FEASIBILITY

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project.

ECONOMIC FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

COST BASED STUDY

It is important to identify cost and benefit factors, which can be categorized as follows:

- Development costs; and
- Operating costs.

This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system.

TIME BASED STUDY

This is an analysis of the time required to achieve a return on investments. The benefits derived from the system. The future value of a project is also a factor. As per the cost-based study this system requires the designing and implementing environment as listed below

LEGAL FEASIBILITY

Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

This system satisfies all the legal requirements and it also complying with the local data protection act.

OPERATIONAL FEASIBILITY

Is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development?

This system operates well in the running environment and run as per the definition provided in the system definition.

SCHEDULE FEASIBILITY

A project will fail if it takes too long to be completed before it is useful. Typically, this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines.

CHAPTER 9

SYSTEM DESIGN

INTRODUCTION

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

PROJECT EXECUTIVE SUMMARY

This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. If appropriate, include the information discussed in the subsequent sections in the summary.

SYSTEM OVERVIEW

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements trace ability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

DESIGN CONSTRAINTS

This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design.

SOFTWARE DETAILED DESIGN

A software module is the lowest level of design granularity in the system. Depending on the software development approach, there may be one or more modules per system. This section should provide enough detailed information about logic and data necessary to completely write source code for all modules in the system (and/or integrate COTS software programs).

If there are many modules or if the documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each module, its functionality, and its hierarchy. Industry-standard module specification practices should be followed. Include the following information in the detailed module designs:

A narrative description of each module, its function(s), the conditions under which it is used (called or scheduled for execution), its overall processing, logic, interfaces to other modules, interfaces to external systems, security requirements, etc.; explain any algorithms used by the module in detail

For COTS packages, specify any call routines or bridging programs to integrate the package with the system and/or other COTS packages (for example, Dynamic Link Libraries) Data elements, record structures, and file structures associated with module input and output

Graphical representation of the module processing, logic, flow of control, and algorithms, using an accepted diagramming approach (for example, structure charts, action diagrams, flowcharts, etc.)

Data entry and data output graphics; define or reference associated data elements; if the project is large and complex or if the detailed module designs will be incorporated into a separate document, then it may be appropriate to repeat the screen information in this section.

INPUT DESIGN

Input Screen must be design in such a way to give an easy navigation throughout the screen without the violation of the input validation. Input design is the process of converting the user-originated data into a computer-based format. Inaccurate input data are the most common cause of error in data processing. The goal of an input data is collected and organized into a group and error free. Input data are collected and organized into a group of similar data. Once identified, appropriated input media are selected for processing. The design was done with six major objectives in mind

- Effectiveness
- Accuracy
- Ease of Use
- Consistency
- Simplicity
- Attractiveness

The main objective of designing input focuses on:

- Controlling the amount of input required
- Avoiding delayed response
- Controlling errors
- Keeping process simple
- Avoiding errors
- Producing cost effective method of input.
- Achieving highest possible level of accuracy.

Ensure that the input is acceptable to and understood by the staff.

The goal of designing input data is to make entry easy, logical and free from errors as possible. The entering data entry operators need to know the allocated space for each field, field sequence and which must match with that in the source

document. The processor analyses the input required. It is then accepted or rejected.

OUTPUT DESIGN

The normal procedure in developing a system is to design the output in detail first and then move back to the input. The output will be in the form of views and reports. The output from the system is required to communicate the result of processing to the users. They are also used as the permanent copy for later verifications.

OUTPUT DESIGN CONSIDERATION

The purpose of outputs has been understood and the efficiency of information contained should be analysed and confirmed. Then the output has been defined in terms of

- Name of the Output
- Content
- Format
- Frequency

Outputs

This section describes of the system output design relative to the user/operator; show a mapping to the high-level data flows described in Section. System outputs include reports, data display screens and GUIs, query results, etc. The output files are described in Section 3 and may be referenced in this section. The following should be provided, if appropriate:

- Identification of codes and names for reports and data display screens

- Description of report and screen contents (provide a graphic representation of each layout and define all data elements associated with the layout or reference the data dictionary)
- Description of the purpose of the output, including identification of the primary users
- Report distribution requirements, if any (include frequency for periodic reports)
- Description of any access restrictions or security considerations

CODE DESIGN

A design pattern is a standardized solution to a software design issue or problem which is encountered daily in real-world application development. A pattern focuses on class design and object interaction. Knowledge of design patterns not only prevents having to re-invent the wheel, it allows developers to discuss their work at a higher level of abstraction.

Design patterns have been the bane of my programming existence. I have trouble learning and remembering them. On the one hand, I feel like I have always been following such patterns throughout my career—even before object-oriented languages. On the other hand, I haven't been able to get a good enough handle on patterns and the terminology to be able chat freely about them with my colleagues

DATABASE DESIGN

The database design involves creation of tables that are represented in physical database as stored medical information. They have their own existence. Each table constitute of rows and columns where each row can be viewed as record that consist of related information and column can be viewed as field of data of same type.

SYSTEM DESIGN FRAMEWORK

The design effort transforms the detailed, defined requirements into complete, detailed specifications that direct development and testing. Design decisions detail how the system will meet the defined functional, physical, interface, security, and data requirements. At the end of the design process the design is base lined.

The general system characteristics are defined during design. The operating system is established and the automated system packaged into major design subsystems. Inputs and outputs of each subsystem are defined, interfaces to external systems are designed, and administrative activities are established. Security and auditing needs are also addressed.

A more detailed structure of the system is then created based on the subsystems identified by the general characteristics. Each subsystem is partitioned into one or more design units, or modules. The process is described in a structure chart, flowchart, action diagram, pseudo code, or other acceptable format for each design unit, or module. Detailed logic specifications are written for each module described and data usage is physically defined to the elemental level. Functions requiring user input and approval are completed in this activity.

Throughout the design phase there are a series of check point and review processes. The design is reviewed to verify that it has the following characteristics:

- Is directly traceable to the requirements.
- Describes how the capabilities defined by the requirements will be implemented.
- The SDD includes
- User, human computer interface design

- System architecture
- Detailed system design
- Data base design including a physical data model and data dictionary.

CHAPTER 9

PROGRAMMING

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.models import Sequential
import numpy as np
from numpy.linalg import norm
import os
from tqdm import tqdm
import pickle
model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
model.trainable = False
model = Sequential([model, GlobalMaxPooling2D()])
#model.summary()
def extract_features(img_path,model):
img = image.load_img(img_path,target_size=(224,224))
img_array = image.img_to_array(img)
expand_img = np.expand_dims(img_array,axis=0)
preprocessed_img = preprocess_input(expand_img)
result_to_resnet = model.predict(preprocessed_img)
flatten_result = result_to_resnet.flatten()
```

```
# normalizing
result_normlized = flatten_result / norm(flatten_result)
return result_normlized
#print(os.listdir('fashion_small/images'))
img_files = []
for fashion_images in os.listdir('fashion_small/images'):
images_path = os.path.join('fashion_small/images', fashion_images)
img_files.append(images_path)
# extracting image features
image_features = []
for files in tqdm(img_files):
features_list = extract_features(files, model)
image_features.append(features_list)
pickle.dump(image_features, open("image_features_embedding.pkl", "wb"))
pickle.dump(img_files, open("img_files.pkl", "wb"))
```

TENSORFLOW:

```
import streamlit as st
import tensorflow
import pandas as pd
from PIL import Image
import pickle
import numpy as np
from tensorflow.keras.preprocessing import image
```

```
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.models import Sequential
from numpy.linalg import norm
from sklearn.neighbors import NearestNeighbors
import os
features_list = pickle.load(open("image_features_embedding.pkl", "rb"))
img_files_list = pickle.load(open("img_files.pkl", "rb"))
model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
model.trainable = False
model = Sequential([model, GlobalMaxPooling2D()])
st.title('Clothing recommender system')
def save_file(uploaded_file):
try:
with open(os.path.join("uploader", uploaded_file.name), 'wb') as f:
f.write(uploaded_file.getbuffer())
return 1
    except:
        return 0
def extract_img_features(img_path, model):
img = image.load_img(img_path, target_size=(224, 224))
img_array = image.img_to_array(img)
```

```
expand_img = np.expand_dims(img_array, axis=0)
preprocessed_img = preprocess_input(expand_img)
result_to_resnet = model.predict(preprocessed_img)
flatten_result = result_to_resnet.flatten()
# normalizing
result_normlized = flatten_result / norm(flatten_result)
return result_normlized
def recommendd(features, features_list):
neighbors = NearestNeighbors(n_neighbors=6, algorithm='brute',
metric='euclidean')
neighbors.fit(features_list)
distence, indices = neighbors.kneighbors([features])
return indices
uploaded_file = st.file_uploader("Choose your image")
if uploaded_file is not None:
if save_file(uploaded_file):
show_images = Image.open(uploaded_file)
size = (400, 400)
resized_im = show_images.resize(size)
st.image(resized_im)
```

 # Extract Features of Uploaded Image

```
features = extract_img_features(os.path.join("uploader", uploaded_file.name),
model)
#st.text(features)
```

```
img_indicess = recommendd(features, features_list)
col1,col2,col3,col4,col5 = st.columns(5)
with col1:
st.header("I")
st.image(img_files_list[img_indicess[0][0]])
with col2:
st.header("II")
st.image(img_files_list[img_indicess[0][1]])
with col3:
st.header("III")
st.image(img_files_list[img_indicess[0][2]])
with col4:
st.header("IV")
st.image(img_files_list[img_indicess[0][3]])
with col5:
st.header("V")
st.image(img_files_list[img_indicess[0][4]])
else:
st.header("Some error occur")
```

TESTING:

```
import pickle
import numpy as np
```

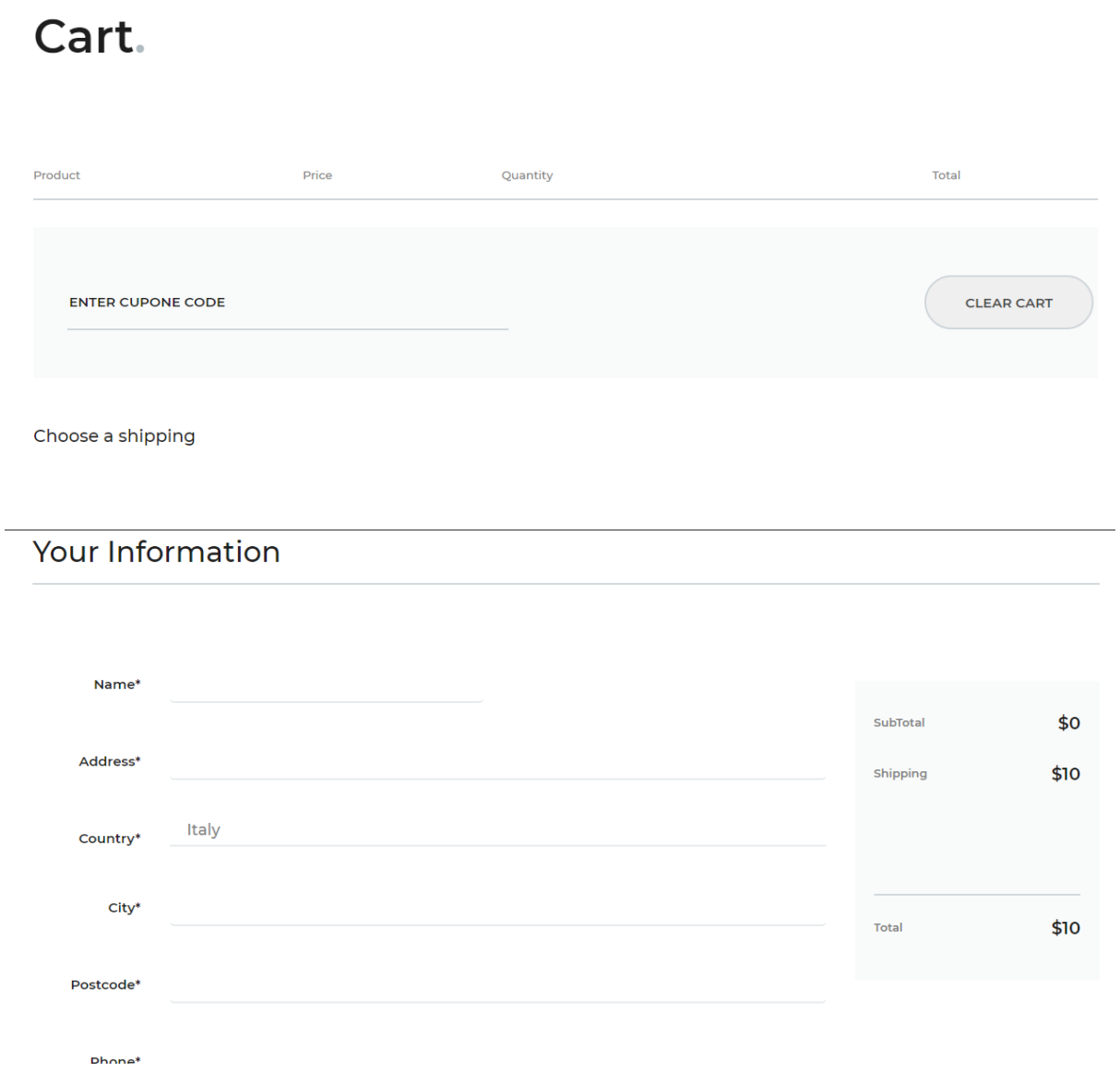
```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.models import Sequential
from numpy.linalg import norm
from sklearn.neighbors import NearestNeighbors
import cv2
features_list = pickle.load(open("image_features_embedding.pkl", "rb"))
img_files_list = pickle.load(open("img_files.pkl", "rb"))
print(np.array(features_list).shape)
model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
model.trainable = False
model = Sequential([model, GlobalMaxPooling2D()])
img = image.load_img('sample/shoes.jpg',target_size=(224,224))
img_array = image.img_to_array(img)
expand_img = np.expand_dims(img_array,axis=0)
preprocessed_img = preprocess_input(expand_img)
result_to_resnet = model.predict(preprocessed_img)
flatten_result = result_to_resnet.flatten()
# normalizing
result_normlized = flatten_result / norm(flatten_result)
neighbors = NearestNeighbors(n_neighbors = 6, algorithm='brute',
metric='euclidean')
```

```
neighbors.fit(features_list)
distence, indices = neighbors.kneighbors([result_normlized])
print(indices)
for file in indices[0][1:6]:
print(img_files_list[file])
tmp_img = cv2.imread(img_files_list[file])
tmp_img = cv2.resize(tmp_img,(200,200))
cv2.imshow("output", tmp_img)
cv2.waitKey(0)
```

CHAPTER 10

RESULT AND DISCUSSION

Users of this web app may locate goods using images. By clicking a photograph of the product and uploading it to the Web App, it offers a simple interface that helps the customer select a product that they may find aesthetically appealing. Products that the model determined to be similar to the uploaded image will be shown to the user. If consumers want to do so, they can purchase the product via the relevant e-commerce websites.



Login.

Your Information

Email

Password

Remember your password?

LOGIN

Related Products

photo of perfume

white labeled perfume bottle

perfume bottle on a book

clear fragrance bottle

CHAPTER 11

CONCLUSION AND FUTURE WORK

Algorithms that provide product suggestions are the best at giving clients a better user experience. A product suggestions engine can assist in bringing clients the pertinent items they desire or need using machine learning, manual curation, and certain algorithms. It enables marketers to instantly provide clients pertinent product recommendations. Product suggestions are used as part of an e-commerce customization strategy to dynamically add goods to websites, apps, contact centers, or emails, improving the user experience.

The greatest algorithms for improving user experiences for customers are those that provide product recommendations. Using machine learning, manual curation, and specific algorithms, a product recommendations engine may help provide customers the relevant products they want or need. It enables marketers to provide clients relevant product suggestions right away.

The present paper presents the development of a system that recognizes fashion similar images. We accomplish this by implementing an already existing CNN model with transfer learning for cloth image recognition using different libraries. For this purpose, we created a plan for collecting data and for developing the steps needed for preprocessing and cleaning up the data. We took into account features like patterns, machine, fabric, style etc. After extensive preprocessing and cleaning of data in a dataset, we constructed the model of stacked CNN to predict the features specific to these attributes and to train the models with the dataset to generate accurate predictions regarding almost all forms of images. A stacked CNN was used and implemented, with the help of this algorithm through which the system can recommend similar images. This is the last test to assess if deep learning for style recovery is at a high development and can be utilized in making fashion choices.

REFERENCES

1. M Sridevi, N ManikyaArun, M Sheshikala and Sudarshan E (2020):" Personalized fashion recommender system with image based neural networks" IOP Conf. Series: Materials Science and Engineering 981 (2020) 022073 IOP Publishing doi:10.1088/1757-899X/981/2/022073.
2. Anjan M, Abhishek V, C. Balamanikantan, Dheeraj, Dr. Venugeetha Y (2022):" Fashion recommendation system using CNN", International Journal of Advance Research, Ideas and Innovations in Technology, ISSN: 2454-132X Impact factor: 6.078 (Volume 6, Issue 3).
3. Muhammad KHALID, Mao KEMING, Tariq HUSSAIN (2021)," Design and implementation of clothing fashion style recommendation system using deep learning", Revista Română de Informatică și Automatică, vol. 31, nr. 4, 123-136, 2021.
4. Batuhan AS¸ IROGLU, Batuhan AS¸ IROGLU, Alkan BALKAYA, Erden TUZ¨ UNKAN, Mustafa Dagtekin (2019)," Smart Clothing Recommendation System with Deep Learning", 978-1-7281-3789-6/19/$31.00 © 2019 IEEE.
5. Narges Yarahmadi Gharaei, Chitra Dadkhah, Lorence Daryoush (2021)," Content-based Clothing Recommender System using Deep Neural Network", 26th International Computer Conference, Computer Society of Iran, Tehran, Iran, 978-1-6654-1241-4/21/$31.00 ©2021 IEEE.
6. Illa Pavan Kumar, Swathi Sambangi (2019)," Content Based Apparel Recommendation System for Fashion Industry", Blue Eyes Intelligence Engineering & Sciences Publication,Retrieval Number F7880088619/2019©BEIESP DOI: 10.35940/ijeat.F7880.088619 .
7. Samit Chakraborty, Md. Saiful Hoque, Naimur Rahman Jeem, Manik Chandra Biswa, Deepayan Bardhan and Edgar Lobaton (2021)," Fashion Recommendation Systems, Models and Methods: A Review", Informatics 2021, 8, 49. https://doi.org/10.3390/informatics8030049.

8. Reshma B Nair , N Susila , Ranjeetha Priya K , Sruthi Anand (2018):" A Flask based Application for Recommending the Similar Kind of Dresses", International Journal of Engineering & Technology, 7 (4.19) (2018) 319-322.
9. Akshit Tayade, Vidhi Sejpal, Ankit Khivasara (2021),"Deep Learning Based Product Recommendation System and its Applications" International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072.
10. Gursimran Kaur, Hrithik Malhotra, Tanmaya Gupta (2021),"Choosing the Right Model: A Comprehensive Analysis of Outfit Recommendation Systems ",International Journal of Computer Applications (0975 - 8887) Volume 183 - No.12, June 2021.
11. Seema Wazarkar , Shruti Patil, Pratik S. Gupta , Kriti Singh , Mukund Khandelwal , C.V. Sri Vaishnavi , Ketan Kotecha (2022)," Advanced Fashion Recommendation System for Different Body Types using Deep Learning Models", DOI: https://doi.org/10.21203/rs.3.rs-1856954/v1.