Purbanchal University

College of Information Technology & Engineering

Subidhanagar, Tinkune, Kathmandu

Project Report

On

Fashion Products Recommender

Submitted by

Sujan Thapa (360237)

Raman Chaudhary (360230)

Anish Chaudhary (360221)

Batch: 2020

Faculty: Science & Technology

Program: BIT

Submitted to

Department of science and technology

Date of Submission: November 25, 2024

Date: November 25, 2024

Letter of Approval

We, the supervising committee of CITE College, have successfully supervised and Approved the Project report entitled “Fashion Product Recommender” submitted by the Project members (Sujan Thapa, Raman Chaudhary & Anish Chaudhary), BIT-VI Semester. During our supervising period, we found that the corresponding report has been prepared as approved by the department in prescribed format of Bachelor of Information Technology (BIT), Faculty of Science & Technology. This report is forwarded to the further Examination.

With best regards,

Project Supervisor

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

With immense please, we are presenting “Fashion Product’s Recommender (FPR)” report as part of curriculum “Bachelor of Information Technology”. We wish to thank all the people who gave us unending support.

We express our profound thanks to our Academic Director Mr. Saroj Dahal, Project Supervisor Mr. Dipraj Kayastha & other’s all those who have indirectly guided and helped us to build this project.

We would like to thank Purbanchal University (PU) as well as College of Information Technology & Engineering (CITE) for the course and the project assigned which helped in our study and developing skills like coding, project management, machine learning, AI, etc.

ABSTRACT

The Fashion Products Recommender is a web application that utilizes advanced machine learning and computer vision techniques to provide personalized fashion recommendations. Built on the Django framework, the system offers a user-friendly interface where users can upload an image of a clothing item. The system then leverages a pre-trained ResNet50 model to extract visual features from the uploaded image. By comparing these features with a database of pre-computed fashion item features using the Nearest Neighbors algorithm and Euclidean distance, the system identifies and displays visually similar fashion items. To enhance the user experience, the system incorporates a degree of randomness in the recommendations. This project demonstrates the successful integration of Machine learning and web development technologies to create a practical and innovative fashion recommendation solution.

[*Keywords: Fashion Product Recommender, Machine Learning, ResNet50, K-Nearest Neighbors, Image-Based Recommendations, Django Framework, Visual Feature Extraction, Fashion Product Dataset, Euclidean Distance, Deep Learning, Feature Normalization, Streamlit Prototype, Algorithms*]

Contents

Chapter 1: Introduction 1

1.1 Background 1

1.2 Introduction to Fashion Products Recommender 1

1.3 Problem Statement 2

1.4 Objectives 3

1.5 Motivation 3

1.6 Applications 5

1.7 Scope 5

1.8 Feasibility Study 6

1.8.1 Technical Feasibility 6

1.8.2 Economic Feasibility 6

1.8.3 Operational Feasibility 7

Chapter 2: Literature Review 8

2.1 Study on Existing Systems 8

2.2 Relevance of Literature Review with project 10

Chapter 3: Design Methodology 11

3.1 Prototype Model 11

Chapter 4: Implementation 14

4.1 Environment Setup 14

4.1.1 Hardware Requirements 14

4.1.2 Software Requirements 15

4.1.3 Development Environment 15

4.2 Dataset Description 15

4.2.1 Dataset Source 15

4.2.2 Dataset Details 16
4.2.3 Data Preprocessing Steps 16
4.3 Model Selection 18
4.3.1 Algorithm Chosen 18
4.4 Implementation workflow 20
4.4.1 Steps in Model Building 20
4.4.2 Prototype Phase 21
4.4.3 Final Project Deployment 21
4.5 Challenges Faced 22
4.6 Block Diagram 22
4.7 Flowchart 23
4.8 Team Structure 24
4.9 Implementation Details 24
4.10 Gantt Chart 25
4.11 Testing of System 26
4.11.1 White Box Testing 26
4.11.2 Black Box Testing 28
Chapter 5: Conclusion 29
5.1 Limitations 29
5.2 Future Enhancement 30
References 31
Appendix 33

List of Tables

Table 1: Team Structure... 24

Table 2: Implementation Details... 24

List of Figures

Figure 1: Prototype model 12

Figure 2: Block Diagram 22

Figure 3: Flow Chart... 23

Figure 4: Gantt Chart 25

Figure 5: Image Feature Extraction 26

Figure 6: Result of Image feature extraction 26

Figure 7: feature dump function 26

Figure 8: Trained Models.. 27

Figure 9: Image save function... 27

Figure 10: Saved uploaded images 27

Figure 11: Empty file upload... 28

Figure 12: File Uploaded 28

Figure 13: Final Result.. 28

Figure 14: Home Page 33

Figure 15: Recommendation Result.. 33

List of Abbreviations

AI: Artificial Intelligence

CNN: Convolutional Neural Network

CSS: Cascading Style Sheets

FPR: Fashion Products Recommender

GPU: Graphics Processing Unit

HDD: Hard Disk Drive

IDE: Integrated Development Environment

KNN (k-NN): k-Nearest Neighbors

L2 Norm: Euclidean Norm for Normalization

RAM: Random Access Memory

SSD: Solid State Drive

VS Code: Visual Studio Code

IDE: Integrated Development Environment

CHAPTER 1: INTRODUCTION

1.1 Background

In the online internet era, the idea of Recommendation technology was initially introduced in the mid-90s. Proposed CRESA that combined visual features, textual attributes and visual attention of the user to build the clothes profile and generate recommendations. Utilized fashion magazines photographs to generate recommendations. Multiple features from the images were extracted to learn the contents like fabric, collar, sleeves, etc., to produce recommendations. In order to meet the diverse needs of different users, an intelligent Fashion recommender system is studied based on the principles of fashion and aesthetics. To generate garment recommendations, customer ratings and clothing were utilized in the history of clothes and accessories, weather conditions were considered in to generate recommendations (Sawalkar, Udar, Bobade, Gupta, & Kahar, 2014).

1.2 Introduction to Fashion Products Recommender

Recommendation systems play a vital role for several online websites, applications and e-commerce services, like social-network, recommendation of items and entities like movies, music, clothes, books and articles. People in today's world have a pool of choices to choose from. It can be the choices of movies, songs, books, clothes, etc. Also there has been an information explosion in the past few years. There has been a gradual increase in the amount of available digital information, electronic data in recent years. This information explosion makes it hard for people to manage their time. In order to help the people to come up with the humongous number of choices, recommendation systems have been developed and have been put to work. Recommendation systems can help the people know what is right by making available the things they will like easily based upon their preferences and likings. Performance of recommendation systems have a significant influence on the success of the commercial companies and industries on the basis of revenue generation and their customer satisfaction. Recommendation systems are distinct from other types of expert systems because these integrate an expert's subject knowledge with the customer’s or the user's preferences to filter out the available data and information for a particular user.

Machine learning is a critical component for a recommendation system. It is a method for giving computers intelligence by simulating how the human brain works. It has been extensively used in the data mining and knowledge discovery fields. The two very basic approaches used to create the recommendation systems are the content-based filtering technique and the collaborative filtering technique.

With an increase in the standard of living, peoples' attention gradually moved towards fashion that is concerned to be a popular aesthetic expression. Humans are inevitably drawn towards something that is visually more attractive. This tendency of humans has led to the development of the fashion industry over the course of time. However, given too many options of garments on the e-commerce websites, has presented new challenges to the customers in identifying their correct outfit. Thus, in this project, 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 the 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 Fashion Product Images Dataset and the Nearest neighbors backed recommender to generate the final recommendations.

1.3 Problem Statement

Humans are inevitably drawn towards something that is visually more attractive. This tendency of humans has led to development of fashion industry over the course of time. With introduction of recommender systems in multiple domains, retail industries are coming forward with investments in latest technology to improve their business. Fashion has been in existence since centuries and will be prevalent in the coming days as well. Women are more correlated with fashion and style, and they have a larger product base to deal with making it difficult to take decisions. It has become an important aspect of life for modern families since a person is more often than not judged based on his attire. Moreover, apparel providers need their customers to explore their entire product line so they can choose what they like the most which is not possible by simply going into a cloth store. The main focus of this project was to build an integrated system that can recommend products which user is mostly likely to choose for themselves.

1.4 Objectives

The objectives which we want to achieve through this project are given below:

- To build a model using effective algorithm which recommends the products for users based on the similar users’ interests.
- To design a system that leverages visual features from input image and recommend fashion products with similar styles, enhancing the accuracy of image-based recommendations.

1.5 Motivation

The Fashion Products Recommender (FPR) is designed to address the growing demand for personalized and seamless online shopping experiences in the fashion industry. With the rapid expansion of e-commerce, customers are often overwhelmed by the vast number of available products, making it challenging to discover items that align with their preferences efficiently.

Before developing this project, we identified significant challenges faced by both customers and businesses. From the customer's perspective, the lack of personalized recommendations often results in frustration and reduced satisfaction during the shopping process. On the other hand, businesses face difficulties in engaging users, converting visits into purchases, and managing inventory effectively without insights into customer preferences.

Traditional methods of product recommendations, such as static filtering or manual curation, are not only time-consuming but also fail to address the dynamic and diverse needs of users. This gap inspired the development of an intelligent recommendation system powered by machine learning, specifically the KNN algorithm, which ensures that users receive relevant product suggestions tailored to their tastes.

The primary motivation for this project stems from the following factors:

- **Enhancing Customer Experience:** The system aims to simplify the shopping journey by presenting users with products they are likely to prefer, thus saving time and improving satisfaction.
- **Driving Business Growth:** By boosting engagement and increasing conversion rates, the system helps businesses achieve higher revenue while fostering customer loyalty.
- **Efficiency in Decision-Making:** The system automates the recommendation process, reducing the reliance on manual efforts and enabling businesses to focus on other strategic priorities.
- **Addressing Industry Needs:** The fast-paced nature of the fashion industry requires tools that can quickly adapt to new trends and customer behaviors, making the FPR an essential solution.
- **Leveraging Advanced Technologies:** The project provides an opportunity to explore and implement machine learning techniques, highlighting the practical applications of algorithms like KNN in solving real-world problems.

Through this project, we aim to develop an efficient, scalable, and cost-effective solution that bridges the gap between customer expectations and business offerings. The motivation lies in creating a system that not only benefits users but also empowers businesses to thrive in the competitive e-commerce landscape.

1.6 Applications

- **E-commerce Platforms:** Enhances user experience by providing personalized and visually similar product recommendations, streamlining the shopping process.
- **Fashion Design and Retail:** Assists designers and retailers in understanding trends by analyzing patterns in user preferences and visual similarities.
- **Inventory Management:** Helps businesses identify popular styles and manage stock based on customer preferences and recommendation trends.
- **Virtual Styling Tools:** Powers applications that allow users to upload an image and find matching or complementary fashion items for a cohesive outfit.

1.7 Scope

The Fashion Products Recommender (FPR) aims to revolutionize the way customers interact with online fashion retailers by providing personalized and efficient product suggestions. The scope of the system is vast, with applications in enhancing user engagement, improving sales performance, and fostering customer loyalty. Some of the scope areas are as follows:

- **Image-Based Recommendations:** Allow users to upload an image, and the system recommends similar or complementary products. Useful for replicating looks or finding matching accessories.
- **Visual Search Integration:** Let users find similar fashion items using a photo or screenshot. Enhance the shopping experience by enabling users to locate products they see online or in real life.

1.8 Feasibility Study

A feasibility study is an assessment of the practicality of a project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained (Simpli Learn, 2024).

From all the study done regarding the feasibility of the proposed system, it can be said that the system was feasible to develop. Feasibility study on the project can be categorized in the following:

1.8.1 Technical Feasibility

Technical feasibility is the process of figuring out how you're going to produce your product or service to determine whether it's possible for your company. Before launching your offerings, you must plan every part of your operations, from first sourcing your production materials all the way to tracking your sales (Indeed, 2024).

The FPR leverages machine learning, specifically the KNN algorithm, which is well-suited for recommendation systems due to its simplicity and effectiveness. Modern computational tools, such as Python and its libraries (e.g., scikit-learn, pandas, and NumPy), provide the necessary framework for developing and deploying the system which are easily available to use.

1.8.2 Economic Feasibility

Economic feasibility refers to the ability of a project or business venture to generate enough revenue to cover its costs and provide a reasonable return on investment. It involves analyzing the costs and benefits of a project, including the costs of materials, labor, and equipment, as well as the projected revenue from sales or other sources of income (Digits Mark, 2024).

Determining the financial resources required for acquiring, implementing, and maintaining the FPR. Here, we have estimated costs related to software licenses, hardware upgrades, customization, and ongoing support. We have also compared these costs with potential savings and benefits to ensure a that our system is cost effective or not.

1.8.3 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves 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 (Law Insider, 2024).

The system is designed to be user-friendly, ensuring seamless integration into existing e-commerce platforms. The FPR provides both customer-facing and admin-facing interfaces, making it intuitive for end users and business administrators alike. Minimal training is required for staff to manage and analyze the recommendation insights. This operational ease makes the system practical for real-world application.

CHAPTER 2: LITERATURE REVIEW

2.1 Study on Existing Systems

To understand the current landscape of fashion product recommendation systems, several existing systems and methodologies were reviewed. The study involved analyzing various documentation and research papers related to product recommendations in e-commerce, particularly focusing on personalization algorithms and the integration of machine learning models. Below are some of the relevant systems and studies examined:

Daraz, a prominent e-commerce platform in Nepal, provides a tailored shopping experience by leveraging various recommendation mechanisms. Its recommendation system focuses on user behavior, browsing history, and purchasing patterns to suggest relevant products. For instance, Daraz highlights complementary items and trending products, enhancing customer engagement and satisfaction. Moreover, its integration with DarazMall ensures that users receive authentic and quality-assured products, fostering trust and repeat usage (Daraz, 2024) .

One of the key technologies explored in fashion product recommendations is **Adobe Commerce**, which offers a comprehensive product recommendation service. The system integrates customer behavior data and product attributes to deliver personalized product suggestions. Through its sophisticated machine learning algorithms, Adobe Commerce optimizes the recommendations to reflect users' past purchases, preferences, and browsing history, aiming to enhance conversion rates and overall sales. This approach allows for a more targeted shopping experience, enhancing the user’s journey by offering relevant product suggestions in real-time. The system's success lies in its ability to scale and customize the recommendations, thus offering flexibility to fashion retailers, enabling them to showcase popular, trending, or seasonal products. By leveraging customer profiles, Adobe Commerce improves user engagement and maximizes retail opportunities by presenting items that meet the customer's unique needs (Experience League, 2024).

Dynamic Yield is another influential platform that aids in the development of personalized product recommendation systems. It utilizes machine learning techniques to analyze user data, such as browsing behavior, purchase history, and customer demographics. The platform helps fashion retailers generate tailored product recommendations through its advanced algorithms, enhancing the customer experience and driving sales. The guide emphasizes that product recommendations can be tailored based on various strategies, including collaborative filtering, content-based filtering, and hybrid models. Collaborative filtering identifies patterns in customer behavior to suggest products that similar users have liked, while content-based filtering recommends products with attributes similar to those the customer has interacted with. By combining these methods, Dynamic Yield helps create a seamless and personalized shopping experience for customers. The system also highlights the importance of implementing product recommendation algorithms that consider real-time data and are adaptable to the evolving fashion trends, ensuring that customers always see the most relevant options (Dynamic Yield, 2024).

The **Shopify** article highlights the significant role of machine learning and personalization in enhancing ecommerce platforms through effective product recommendations. Shopify utilizes advanced machine learning models to provide personalized recommendations, improving conversion rates and average order values. These systems rely on co-purchase patterns, buyer behavior, and product similarities, helping merchants present more relevant suggestions to their customers. The article also emphasizes the importance of predictive analytics and dynamic filtering to adapt recommendations based on customer preferences, historical purchases, and real-time data. Features such as visual search and typo-tolerant search algorithms make discovering products more intuitive for shoppers, further boosting user satisfaction and sales. In 2022, Shopify merchants leveraging these recommendation tools saw substantial revenue increases, showcasing the effectiveness of personalized recommendations in driving sales and customer retention (Shopify, 2024).

2.2 Relevance of Literature Review with project

All the literature and systems analyzed in this review are similar in their goal of providing effective and personalized product recommendations. Platforms like Daraz, Adobe Commerce, Dynamic Yield, and Shopify leverage machine learning and personalization techniques to enhance user engagement and improve sales. These approaches use user behavior, browsing history, and advanced filtering methods to offer tailored product suggestions, ensuring relevance and satisfaction. Similarly, our project, “Fashion Product Recommender System,” adopts a related methodology by using machine learning to analyze an input image and recommend visually similar fashion products. While the implementation specifics differ, our project utilizes a Django-based framework with an image-centric focus we have incorporated the foundational concepts and techniques observed in the reviewed systems, adapting them to suit the requirements and scale of our application. These insights have informed our project’s design and execution, ensuring it aligns with industry best practices for effective recommendation systems.

CHAPTER 3: DESIGN METHODOLOGY

Design methodology refers to the development of a system or method for a unique situation. Today, the term is most often applied to technological fields in reference to web design, software or information systems design. The key to design methodology is finding the best solution for each design situation, whether it be in industrial design, architecture or technology. Design methodology stresses the use of brainstorming to encourage innovative ideas and collaborative thinking to work through each proposed idea and arrive at the best solution. Meeting the needs and wants of the end user is the most critical concern. Design methodology also employs basic research methods, such as analysis and testing (Learn, 2024).

For our project, as a part of project we have chosen one of the mostly used model to make the small system which is Prototype model.

3.1 Prototype Model

The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built. A prototype is a toy implementation of the system. A prototype usually turns out to be a very crude version of the actual system, possible exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software. In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed (Javat Point, 2024).

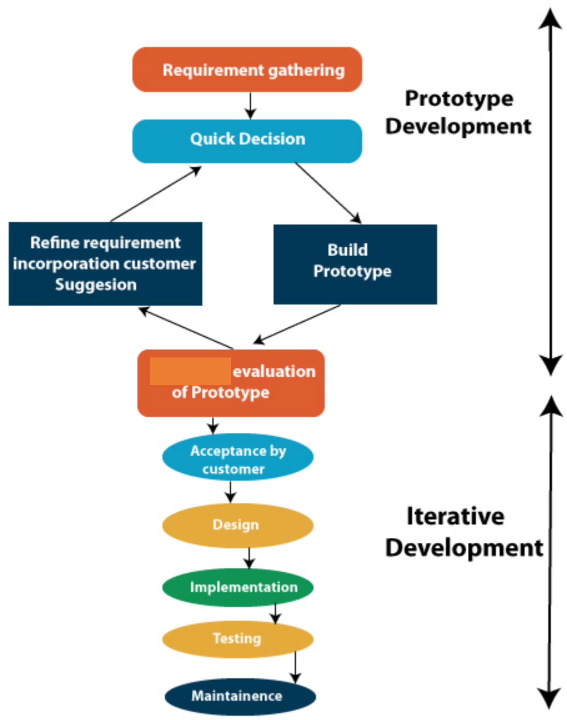


Figure 1: Prototype model

We have used the prototype model for our project, which emphasizes building an initial version of the system, or prototype, to understand and refine the requirements. We chose this model because it allows iterative development, where the prototype is created, evaluated, and improved based on user feedback and testing.

This approach is beneficial for beginners like us as it provides a hands-on learning process and helps us visualize the project's outcome early in development. By repeatedly refining the prototype, we ensured that the final system closely meets the requirements and expectations. This model also helped us identify potential issues and make necessary changes before the final implementation, making the development process more flexible and user-centric.

Once we've selected a project topic, the initial phase involves diligent requirement gathering. This crucial step ensures our prototype accurately reflects the raw version of the project and its core features. Key requirements identified in this phase typically include the model to be used (e.g., Prototype model), programming languages (e.g., Python, HTML, CSS), Integrated Development Environment (IDE) (e.g., VS Code), and essential libraries/modules (e.g., NumPy, scikit-learn, TensorFlow, Streamlit). Additionally, we define the specific functions and features to be incorporated and identify necessary datasets.

With the requirements in place, we embark on the prototype design and development process. Leveraging the gathered tools and technologies, we construct the prototype. Upon completion, a rigorous evaluation is conducted to assess its alignment with the project's specifications.

If the prototype meets the requirements, we transition to the main project development phase, utilizing the Django framework. Once the project is complete, it undergoes thorough implementation and testing to ensure its correct functionality. This phase encompasses various methods, which we'll delve into during the implementation discussion.

Following successful testing, we deploy the project to a web browser to observe its behavior. Ongoing maintenance is essential to address any emerging errors, bugs, or potential feature enhancements.

The project development process begins with rigorous requirement gathering, followed by the creation of a prototype to validate the concept. Once the prototype meets the requirements, the main project is developed using the Django framework. The completed project is then implemented, tested, and deployed to a web browser. Ongoing maintenance is crucial to address any issues and enhance the project's features.

CHAPTER 4: IMPLEMENTATION

Project implementation is the process of putting a project plan into action to produce the deliverables, otherwise known as the products or services, for clients or stakeholders. It takes place after the planning phase, during which a team determines the key objectives for the project, as well as the timeline and budget. Implementation involves coordinating resources and measuring performance to ensure the project remains within its expected scope and budget. It also involves handling any unforeseen issues in a way that keeps a project running smoothly (Indeed, 2024).

It bridges the gap between theoretical design and practical execution. This chapter outlines the steps, tools, and techniques used to develop a robust machine learning model for personalized fashion recommendations. It covers the environment setup, dataset preparation, model architecture, and evaluation methodologies. This documentation provides insights into the technical aspects of the project, highlights challenges faced, and ensures reproducibility, ultimately aligning with the project's objectives.

4.1 Environment Setup

Setting up a development environment is a fundamental step in software development. It involves configuring the software and hardware on which code will be written, tested, and executed. They are given below:

4.1.1 Hardware Requirements

Considering the contemporary computing environment, the hardware requirements for optimal performance are set accordingly. As this project is developed in today's world needs matchable computer so, we have set requirements as per our system developed devices. The hardware requirements for our project are (i3 or higher) level processor because modern computer's support new updated browsers so, to support these browsers functions at the best device must have new functions as we have stated. To support system to run perfectly device must have minimum RAM of 8GB and if available then HDD or SSD of minimum 128GB for the better performance.

4.1.2 Software Requirements

We built the project using Python as the main programming language. Django powered the backend, handling user input and connecting to the machine learning model. Libraries like NumPy, Scikit-learn, and TensorFlow were used for calculations, machine learning, and deep learning. Streamlit helped build an interactive web app. Jupyter Notebook was great for exploring data and training models. Finally, HTML and CSS were used to create a user-friendly interface. All these tools together made our recommender system work.

4.1.3 Development Environment

The project was developed and hosted locally on a Windows 10 or higher operating system. Visual Studio Code (VS Code) was used as the primary IDE for efficient coding and debugging. Jupyter Notebook was utilized for data exploration, model prototyping, and training. This hybrid approach, combining the structured development environment of VS Code with the flexibility of Jupyter Notebook, streamlined the development process.

4.2 Dataset Description

A dataset is a structured collection of data organized and stored together for analysis or processing. The data within a dataset is typically related in some way and taken from a single source or intended for a single project (Data Bricks, 2024).

Some of the fields under dataset description are as follows:

4.2.1 Dataset Source

The dataset for the fashion product recommender system was sourced from Kaggle, a renowned platform that provides a wide range of datasets for machine learning and data science projects. The dataset consisted of image files, which were downloaded and prepared for use in the project. Kaggle’s dataset reliability and diversity made it an ideal choice for creating a system capable of recommending fashion items based on visual features.

4.2.2 Dataset Details

The dataset included over 10,000 image samples representing various categories of fashion products. To prepare the data for machine learning, features were extracted from the images using the ResNet50 model, a pre-trained convolutional neural network. ResNet50 is well-regarded for its ability to capture intricate visual patterns and representations, which proved essential for identifying similarities between different fashion items. The extracted features served as a compact yet informative representation of the images, ensuring efficient and effective processing for the recommender system.

4.2.3 Data Preprocessing Steps

The data preprocessing for the fashion product recommender involved several key steps to prepare the image data for feature extraction and similarity analysis. These steps ensured that the images were properly formatted and normalized for processing by the ResNet50 model. Below is a detailed explanation of the preprocessing steps:

- **Extracting Filenames from the Folder**

The process began by reading all image files from the specified directory (`images`). The filenames were stored in a list to maintain a reference for later retrieval and analysis. This step facilitated organized access to the dataset and prepared it for batch processing.

- **Loading and Preprocessing Images**

Each image was preprocessed before being fed into the ResNet50 model for feature extraction. The following transformations were applied:

Image Resizing: Images were resized to 224x224 pixels, the input size required by the ResNet50 model.

Image Conversion: The image was converted into a numerical array using image.img_to_array() to make it compatible with TensorFlow operations.

Batch Dimension Addition: A new dimension was added using np.expand_dims() to match the input format of the model, which requires a batch-like structure.

Input Preprocessing: The resized image was preprocessed using the preprocess_input() function, which normalized pixel values according to the ResNet50 model's requirements.

- **Feature Extraction Using ResNet50**

The ResNet50 model, pre-trained on the ImageNet dataset, was used to extract high-level features from the images. The top layers of the ResNet50 model were excluded (include_top=False) to focus on feature extraction rather than classification. A GlobalMaxPool2D layer was added to reduce the dimensionality of the extracted features, summarizing the most significant patterns in each feature map. The features were then normalized using the L2 norm (result/norm(result)) to ensure uniform scaling, improving the effectiveness of similarity calculations.

- **Batch Feature Extraction**

A custom function, extract_features_from_images(), was created to automate the preprocessing and feature extraction for each image in the dataset. This function was applied iteratively to all images in the dataset, generating a feature vector for each image.

- **Saving Preprocessed Data**

The extracted features and corresponding filenames were saved as serialized files (Images_features.pkl and filenames.pkl) using Python's pickle module. This step ensured that the preprocessed data could be quickly loaded for future use without reprocessing the entire dataset, saving time and computational resources.

- **Preparing Data for Similarity Analysis**

The normalized feature vectors were used to train a k-Nearest Neighbors (k-NN) model with Euclidean distance as the metric. This model facilitated the retrieval of visually similar images based on the extracted features, enabling effective recommendations.

Thus, these preprocessing steps played a crucial role in transforming raw image data into a structured format suitable for the recommender system, ensuring efficient and accurate similarity analysis.

4.3 Model Selection

This part contains the algorithm we have chosen for the project completion and its detailed explanation:

4.3.1 Algorithm Chosen

The fashion product recommender system utilizes a combination of ResNet50, a pre-trained Convolutional Neural Network (CNN), and the k-Nearest Neighbors (k-NN) algorithm. This hybrid approach effectively extracts features from images and identifies similar products based on those features. Below is the detailed justification for selecting these algorithms:

- **ResNet50 for Feature Extraction**

ResNet-50 is CNN architecture that belongs to the ResNet (Residual Networks) family, a series of models designed to address the challenges associated with training deep neural networks. Developed by researchers at Microsoft Research Asia, ResNet-50 is renowned for its depth and efficiency in image classification tasks (Robo Flow, 2024).

It is used for the project because of the following:

Effectiveness for Image Data: CNNs are specifically designed for image-related tasks, making them ideal for extracting meaningful features from fashion images. ResNet50, a deep CNN architecture, is pre-trained on ImageNet, a large and diverse dataset. This makes it highly effective for capturing intricate patterns, textures, and shapes in images, which are crucial for distinguishing between fashion items.

Transfer Learning: By using the pre-trained ResNet50, the system avoids the computational cost of training a CNN from scratch. The model's lower layers extract general visual features, while its higher layers focus on more complex patterns. This allows the recommender system to leverage ResNet50 as a robust feature extractor.

Dimensionality Reduction: A GlobalMaxPool2D layer was added to the ResNet50 model to summarize and reduce the spatial dimensions of the feature maps. This layer ensures that the extracted feature vectors are compact and informative, simplifying the subsequent similarity computations.

Normalization: The feature vectors were normalized using the L2 norm, ensuring uniform scaling of features, which is essential for distance-based similarity measures.

- **k-Nearest Neighbors (k-NN) for Similarity Matching**

KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data) (Geeks for Geeks, 2024).

It is used because of the following:

Purpose: After extracting feature vectors from the images using ResNet50, the k-Nearest Neighbors (k-NN) algorithm was employed to identify similar images based on their features. The k-NN algorithm is a non-parametric and intuitive method that finds the closest matches to a given query point in feature space.

Distance Metric: Euclidean distance was used as the metric to compute the similarity between feature vectors. This choice was made because Euclidean distance effectively measures the straight-line distance between two points in a multi-dimensional space, which is well-suited for normalized feature vectors.

Number of Neighbors: The parameter `n_neighbors=6` was chosen to retrieve the top 5 most similar images to the query image, in addition to the query image itself. This ensures that users receive multiple recommendations for related fashion products.

By combining ResNet50 and k-NN, the system efficiently extracts meaningful features from images and performs similarity-based retrieval. This two-step approach allows the system to offer accurate and visually relevant recommendations to users, aligning with the goals of a personalized fashion product recommender system. In conclusion, the combination of ResNet50 for feature extraction and k-NN for similarity matching provides a robust and effective approach for identifying and recommending similar fashion items, making it an ideal choice for this project.

4.4 Implementation workflow

In this part of report following fields are explained in detail:

4.4.1 Steps in Model Building

Feature Engineering

- **Feature Extraction**

The ResNet50 model was used to extract high-level feature vectors from all the images in the dataset. Images were resized to 224x224 pixels, preprocessed for compatibility with ResNet50, and passed through the model to generate feature embeddings. These embeddings represented the most critical visual features of each image.

- **Feature Normalization**

The extracted feature vectors were normalized using the L2 norm to ensure uniform scaling, which is essential for distance-based similarity calculations.

Data Splitting

Instead of traditional splitting into training, validation, and testing sets (as used in supervised learning), the recommender system processed the entire dataset for feature extraction. The k-Nearest Neighbors (k-NN) algorithm worked on the extracted feature space, with no explicit need for training-validation splits, as it identifies nearest neighbors on-the-fly during queries.

4.4.2 Prototype Phase

The project prototype was developed using Streamlit, a Python-based framework that enabled the rapid development of an interactive web application. This user-friendly interface allowed users to upload an image and receive visually similar fashion product recommendations based on the k-NN algorithm and ResNet50 features. Streamlit's simplicity and efficiency accelerated the prototyping and testing phases of the project, streamlining the development process.

4.4.3 Final Project Deployment

For the final implementation, the project was transitioned to Django, a robust web framework. Django handled the backend operations, managing user input, processing image uploads, and interfacing with the pre-extracted features and the k-NN model. A visually appealing and interactive user interface was developed using HTML and CSS within Django's templating system. Django's scalability and robust database handling made it suitable for deploying the recommender system as a production-ready application.

4.5 Challenges Faced

One of the significant challenges encountered during the development of this project was the limitation of computational resources. Training a deep learning model like ResNet50 on a large dataset requires substantial computational power, including a powerful GPU. However, due to resource constraints, we had adapted the following approach:

Using Smaller Dataset: We worked with a smaller, more manageable dataset to reduce training time and computational demands. While this approach might have impacted the model's overall performance, it was a necessary compromise given the resource limitations.

4.6 Block Diagram

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

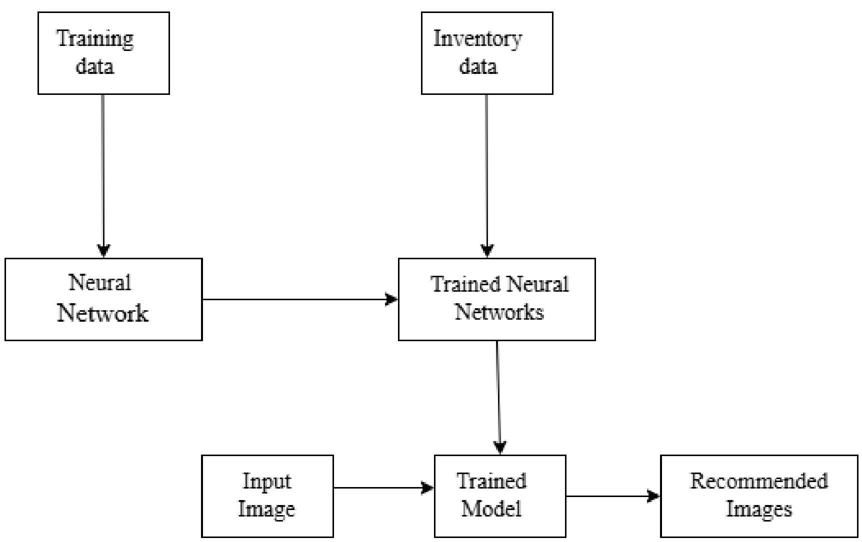


Figure 2: Block Diagram

4.7 Flowchart

A flowchart is a diagram depicting a process, a system or a computer algorithm. It is a diagrammatic representation of the solution to a given problem but, more importantly, it provides a breakdown of the essential steps to solving the problem. When designing and planning a process, flowcharts can help you identify its essential steps and simultaneously offer the bigger picture of the process. It organizes the tasks in chronological order and identify them by type, e.g., process, decision, data, etc. Each step is independent of implementation as the flowchart only describes what should happen at that step, what input is needed and what the output of the step is but it says nothing about how to implement the step (Ucl Ac, 2024).

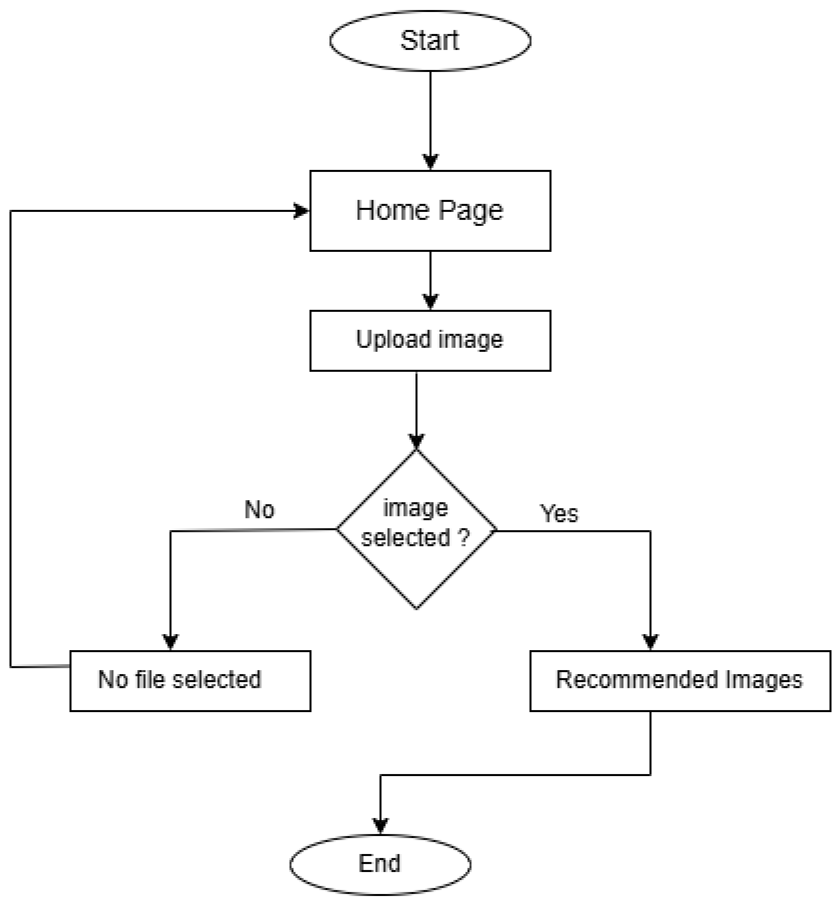


Figure 3: Flow Chart

4.8 Team Structure

Team Members	Symbol no.	Task done
Sujan Thapa	360237	Research, Coding, Debugging & Documentation
Raman Chaudhary	360230	Research, Coding, Debugging & Documentation
Anish Chaudhary	360221	Research, Coding, Debugging & Documentation

Table 1: Team Structure

4.9 Implementation Details

Files	Description
Fashion_Recom_Model.ipynb	This file contains code for initial project stages, including feature extraction, data preprocessing, and model training, etc.
.py files	These are the core files that enable the project to run as a web application, allowing users to interact with it.
.html	This file defines the visual structure and styling of the project's homepage.
.pkl files	These files contain the trained machine learning model using dataset.

Table 2: Implementation Details

4.10 Gantt Chart

A Gantt chart is a project management tool that helps in planning, scheduling and monitoring a project. Using a Gantt chart can improve your planning and scheduling, remote work collaboration, resource allocation and task delegation. A Gantt chart represents all information visually through a horizontal bar graph. Project managers and team members can view the task schedules, dependencies and progress by just glancing at the chart. Planning for all tasks in advance and making them visible in one place empowers teams to deliver on time. Gantt charts make it easy for project managers to identify the critical path to project completion and ensure that there is no delay in those tasks. Project managers should use Gantt charts for project planning and scheduling, allocating resources, tracking the progress of each task at all times and ensuring the smooth and timely execution of critical tasks (Forbes, 2024).

Figure 4: Gantt Chart

Above Gantt chart shows the progress of our project in 6 months period. There we have done project with 5 different sections such as (planning, research & learning, prototype, coding, testing, documentation).

4.11 Testing of System

4.11.1 White Box Testing

Test I: Image features extraction

```
image_features = []
for file in filenames:
    image_features.append(extract_features_from_images(file, model))
image_features
```

Figure 5: Image Feature Extraction

The above code successfully extracted features from all the datasets, preparing them for model training.

Obtained successful result:

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 161ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 203ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 188ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step
```

Figure 6: Result of Image feature extraction

Test II: Dumping the extracted features to train model

```
Image_features = pkl.dump(image_features, open('Images_features.pkl','wb'))

filenames = pkl.dump(filenames, open('filenames.pkl','wb'))
```

Figure 7: feature dump function

The above code successfully trained two machine learning models as shown below:

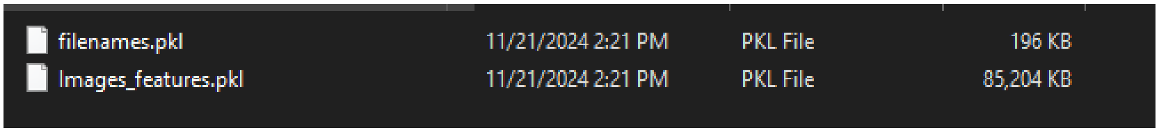
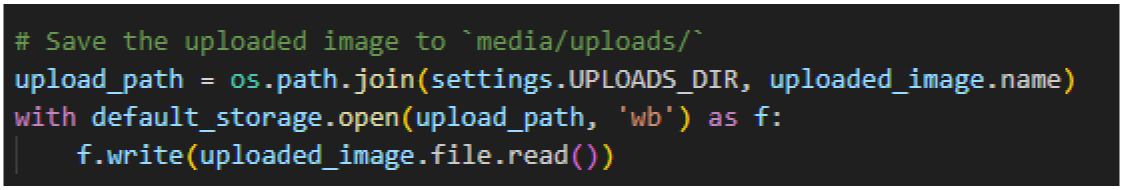


Figure 8: Trained Models

Test III: Uploaded image to be saved in directory



```
# Save the uploaded image to `media/uploads/`
upload_path = os.path.join(settings.UPLOADS_DIR, uploaded_image.name)
with default_storage.open(upload_path, 'wb') as f:
    f.write(uploaded_image.file.read())
```

Figure 9: Image save function

The uploaded image by the user is successfully saved in the uploads directory as follows:

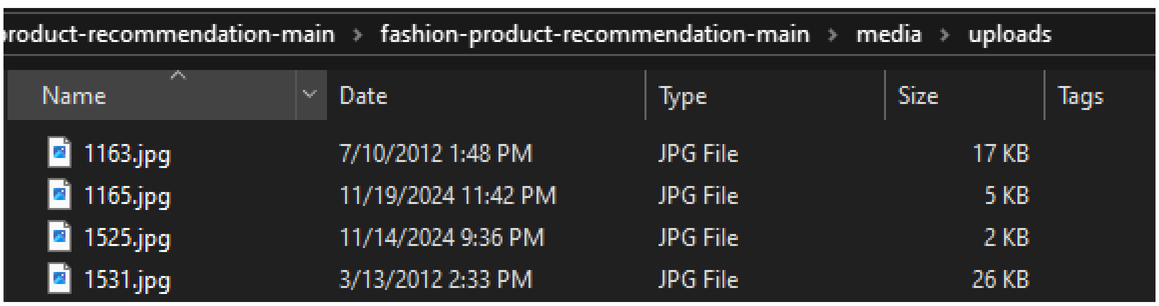


Figure 10: Saved uploaded images

4.11.2 Black Box Testing

Test I: Empty file upload field

If no file is selected and the upload button is pressed, a pop-up message appears indicating "Please select a file".

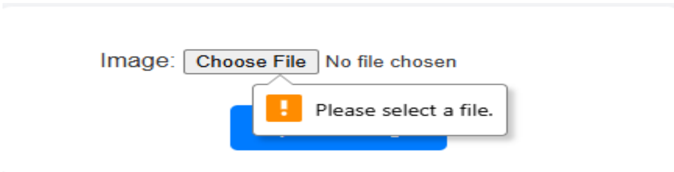


Figure 11: Empty file upload

Test II: File selected and uploaded successfully

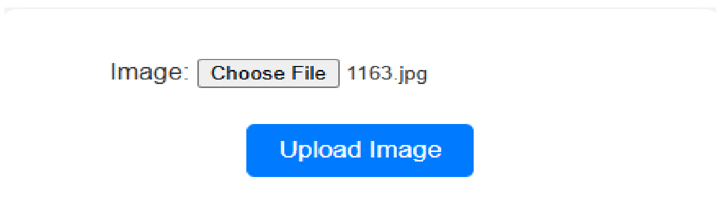


Figure 12: File Uploaded

Once a file is uploaded successfully, the recommended images, along with the uploaded image, are displayed as shown below:

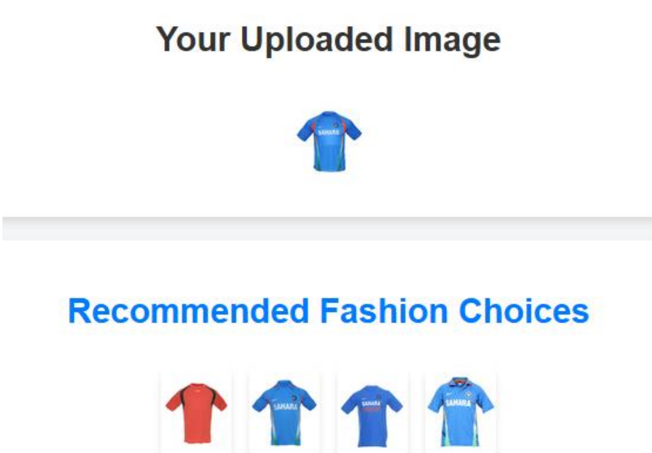


Figure 13: Final Result

CHAPTER 5: CONCLUSION

The Fashion Product Recommender effectively demonstrates the integration of machine learning and web development technologies to address real-world challenges in the fashion industry. By leveraging a pre-trained ResNet50 model for feature extraction and the k-Nearest Neighbors (k-NN) algorithm for similarity matching, the system delivers personalized and visually relevant fashion recommendations. The transition from a Streamlit prototype to a full-fledged Django application highlights the system's scalability and robustness. Through rigorous testing and implementation, the system has proven to be a valuable tool for enhancing user engagement, aiding decision-making, and optimizing inventory management for fashion retailers. The user-friendly interface, efficient feature processing, and scalable architecture contribute to the overall success of the project. This work showcases the potential of combining deep learning and web development technologies to create innovative solutions in the fashion domain.

5.1 Limitations

Any program cannot be 100% reliable and efficient. This program also has some drawbacks which are given below:

- The limited size of the dataset is used to train the model which can potentially impact the accuracy and generalizability of the recommendations.
- The current implementation of the recommender system offers a limited range of filtering options for users.

5.2 Future Enhancement

- As a future direction, we envision expanding the system's capabilities to incorporate a wider range of fashion preferences and styles. This could involve incorporating additional features like personalized recommendations based on user history, social media trends, and seasonal trends. Additionally, integrating a virtual try-on feature could enhance the user experience, allowing users to visualize how different clothing items would look on them.

REFERENCES

Daraz. (2024, 11 10). *Daraz a ecommerce platform.* Retrieved from www.daraz.com.np: https://www.daraz.com.np/

Data Bricks. (2024, 11 6). *What is a Dataset?* Retrieved from www.databricks.com: https://www.databricks.com/glossary/what-is-dataset#:~:text=A%20dataset%20is%20a%20structured,intended%20for%20a%20single%20project.

Digits Mark. (2024, 10 28). *What is the economic feasibility study, its importance, and its types*. Retrieved from www.digitsmark.com/: https://digitsmark.com/en/blogs/what-is-the-economic-feasibility-study-its-importance-and-its-types

Dynamic Yield. (2024, 11 8). *Impactful techniques to help you ace your product recommendations*. Retrieved from www.dynamicyield.com: https://www.dynamicyield.com/lesson/product-recommendations-guide/

Experience League. (2024, 11 11). *Introduction to Product Recommendations*. Retrieved from www.experienceleague.adobe.com: https://experienceleague.adobe.com/en/docs/commerce-merchant-services/product-recommendations/overview

Forbes. (2024, 11 3). *What Is A Gantt Chart? The Ultimate Guide*. Retrieved from www.forbes.com: https://www.forbes.com/advisor/business/software/what-is-a-gantt-chart/

Geeks for Geeks. (2024, 11 1). *K-Nearest Neighbor(KNN) Algorithm*. Retrieved from www.geeksforgeeks.org: https://www.geeksforgeeks.org/k-nearest-neighbours/

Indeed. (2024, 11 5). *The 6 Steps of Project Implementation (With Tips)*. Retrieved from www.indeed.com: https://www.indeed.com/career-advice/career-development/project-implementation

Indeed. (2024, 10 27). *What Is Technical Feasibility? (Process And Best Practices)*. Retrieved from www.in.indeed.com: https://in.indeed.com/career-advice/career-development/what-is-technical-feasibility

Javat Point. (2024, 11 5). *Prototype Model*. Retrieved from www.javatpoint.com: https://www.javatpoint.com/software-engineering-prototype-model

Law Insider. (2024, 11 2). *Operational Feasibility definition*. Retrieved from www.lawinsider.com: https://www.lawinsider.com/dictionary/operational-feasibility

Learn. (2024, 11 5). *What Is Design Methodology?* Retrieved from www.learn.org: https://learn.org/articles/What_is_Design_Methodology.html

Robo Flow. (2024, 11 3). *What is ResNet-50?* Retrieved from www.roboflow.com: https://blog.roboflow.com/what-is-resnet-50/

Sawalkar, P., Udar, B., Bobade, A., Gupta, N., & Kahar, S. (2014). FASHION RECOMMENDATION SYSTEM. *FASHION RECOMMENDATION SYSTEM*, 1-1.

Shopify. (2024, 11 8). *How Ecommerce Product Recommendations Drive Sales*. Retrieved from www.shopify.com: https://www.shopify.com/blog/ecommerce-product-recommendation

Simpli Learn. (2024, 11 1). *What is a Feasibility Study?* Retrieved from www.simplilearn.com: https://www.simplilearn.com/feasibility-study-article#what_is_a_feasibility_study

Ucl Ac. (2024, 11 3). *What is the significance of a flowchart?* Retrieved from www.ucl.ac.uk: https://www.ucl.ac.uk/culture-online/case-studies/2021/jun/what-significance-flowchart

APPENDIX

1) Home Page

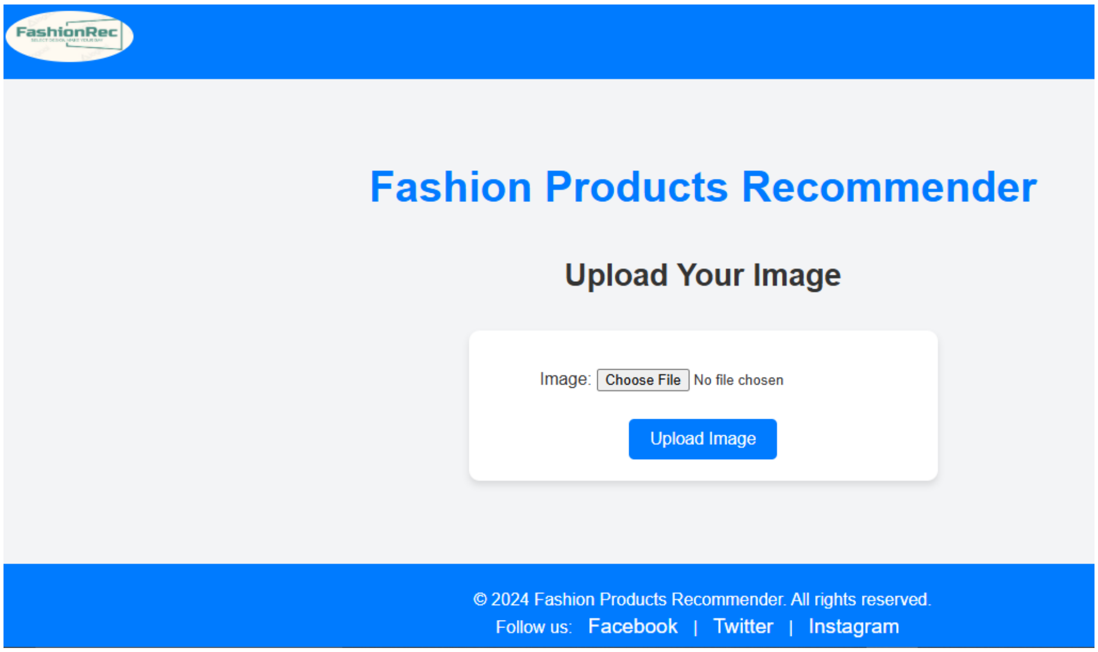


Figure 14: Home Page

2) Recommendation Result

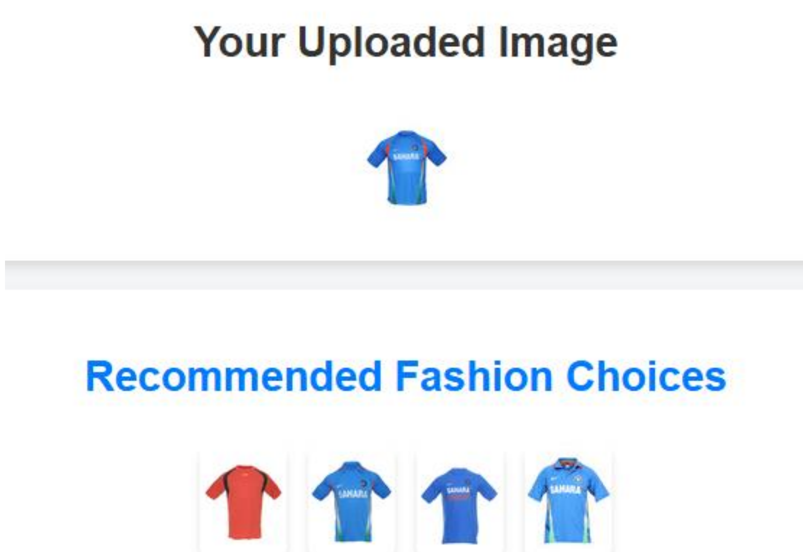


Figure 15: Recommendation Result