CAPSTONE PROJECT REPORT

(Term: January - May 2025)

OUTFITX: A DEEP LEARNING FRAMEWORK FOR PERSONALIZED OUTFIT RECOMMENDATIONS

Submitted by

Satya Reddy Satti	**12104855**
Chanchal Alam	**12106435**
Kriti Mishra	**12106729**

Project Group Number: **K21UG (G15)**

Course Code: CSE461

Under the Guidance of

Mr. Ajay Sharma

School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

OUTFITX: A DEEP LEARNING FRAMEWORK FOR PERSONALIZED OUTFIT RECOMMENDATIONS

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Satya Reddy Satti	**12104855**
Chanchal Alam	**12106435**
Kriti Mishra	**12106729**

Supervisor

AJAY SHARMA

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

May 2025

PAC FORM

PROJECT CONSENT FORM

This is to certify that the project work titled **OutfitX - An Outfit Recommendation System Using Neural Networks** carried out by **Chanchal Alam and Satya Reddy Satti** is submitted to LPU, Phagwara, India in partial fulfilment of the requirements for the **B. Tech (CSE)**. The project work represents an authentic and original effort conducted under Supervisor's guidance.

Student Declaration:
I/We, **Chanchal Alam and Satya Reddy Satti** hereby declare that the project work titled **"OutfitX - An Outfit Recommendation System Using Neural Networks"** is our original work and has not been submitted for any other degree or diploma.

Student's Information:
Name: Chanchal Alam and Satya Reddy Satti , S YOUVRAJ
Roll Number: 12106435,12104855 , 12108274
Signature: S. Satya Reddy
Date: 24-01-2025

Supervisor's Declaration:
I hereby confirm that the project is an authentic submission and has not been used to obtain any other degree or diploma.

Supervisor's Information:
Name: AJAY SHARMA
Signature: [illegible] 63864
Date: 28/APril/2025.

Counter-Signed By:
1) Concerned HOD:
Name: ___________
Signature: _________
Date: __________

DECLARATION STATEMENT

We hereby declare that the project work entitled “OUTFITX: A DEEP LEARNING FRAMEWORK FOR PERSONALIZED OUTFIT RECOMMENDATIONS” is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr. Ajay Sharma, during January to May 2025. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: **K21UG (G15)**

Name of Student 1: **Satya Reddy Satti**
Registration Number: **12104855**

Name of Student 2: **Chanchal Alam**
Registration Number: **12106435**

Name of Student 3: **Kriti Mishra**
Registration Number: **12106729**

(Signature of Student 1)
Date:

(Signature of Student 2)
Date:

(Signature of Student 3)
Date:

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of the B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Designation

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

Date :

ACKNOWLEDGEMENT

We would like to thank the Department of Computer Science and Engineering at Lovely Professional University for providing the lab space and working environment. We want to thank Mr. Ajay Sharma, who was very helpful and always available to clear up our doubts. We feel fortunate to have had their support the entire time. We also want to express our sincere gratitude to Almighty God, our parents, our technical mentors at Upgrad, and our close companions for helping us maintain our motivation throughout the entire dissertation work.

TABLE OF CONTENTS

TABLE OF CONTENTS

CONTENTS	PAGE NO.

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The global fashion industry with revenue exceeding $1.7 trillion is undergoing a digital transformation that stems from the consumers' demands for what I call 'near personalized' customer perception. Even with the development of e-commerce and playful styling tools, customers still encounter several difficulties when building a look from their respective wardrobes. Decision fatigue studies state that 65% of consumers toss outfit purchases over the edge and 30% return items from inadvertently mixed events. Currently, there are existing solutions like StyleSnap from Amazon and Pinnter's visual search, which are based on generic datasets or simple rule-based algorithms (e.g. color matching) and do not consider the preferences, the contextual factors (e.g weather, occasion) or the intricate combined textures, patterns and styles[1].

Traditional recommendation systems focus on popularity over personalization, which is why users end up having no wish-related suggestions. For instance, collaborative filtering cannot handle the "cold start" problem for new users, and the content such as Siamese networks does not deal with pairwise compatibility while never capturing the balanced nonpairwise aspects of holistic cohesion fulfilled context. The users are not aware of why certain items are recommended; these systems are not transparent. Taking fashion as an example, this gap is critical in that subjective preferences, cultural trends, and other unknowns play a much greater role in decisions. To improve this, we recommend OutfitX, a virtual wardrobe system that works with deep learning and image classification to support users when selecting outfits based on uploaded clothing images. [2]. Traditional fashion apps depend on predefined templates, whereas OutfitX uses computer vision, and machine learning to analyze user data in user wardrobes to ensure personalized outfit recommendations. Fashion choices are influenced by social tendencies, personal preferences, and stylist suggestions these days. Digitalization and e-commerce have changed the way individuals approach fashion. AI-controlled recommendation systems can now analyze large amounts of data, identify patterns, and provide users with curated costume selection options that suit their flavors and opportunities. Modern wardrobe. Choosing clothes for everyday life is difficult as many people have a wealth of clothing. [3]. OutfitX solves this problem by implementing an intuitive system that simplifies the decision process. By using deep learning techniques, the system classifies clothing, evaluates style compatibility, and generates costume recommendations that improve fashion adjustments. Proper outfits should take into consideration factors such as color harmony, fabric structure, seasonality, and personal style.

OutfitX integrates several AI-controlled techniques, such as content-based filters and deep learning models, to ensure that users receive high-quality costume suggestions. The system, along with color adjustments, also considers additional fashion trends, occasional clothing, and climate suitability. The ability to digitize your wardrobe and maintain it with AI operation recommendations will improve your general dressing experience. Users can manage their clothing inventory, turn to their clothing, and easily explore new fashion trends. The development of a digital fashion platform combined with AI and augmented reality (AR) has opened up new opportunities for recommendations for interactive and dynamic fashion modes. If you want to improve the accuracy of costume recommendations. The folding network (CNN) distinguishes clothing categories and is successfully identified by fashion elements that are perceived as unique [4].

OutfitX uses a finely tuned ResNet50 model designed with a wide range of data records to ensure a very accurate classification of clothing. OutfitX is different from other rule-based recommendation systems that depend on pre-defined fashion templates as OutfitX continuously learns from user interaction and feedback. As it goes through time, the system fine-tunes its recommendations to become more tailored, more personalized, and better aligned with an individual’s style. Through this adaptive learning, this approach improves user engagement and satisfaction in the process of getting recommendations. OutfitX fills the gap between e-commerce and sustainable fashion. As people become more conscious about fashion waste and sustainability, many users want to find out how to extend the life of their current wardrobe items as much as possible. OutfitX provides outfit combinations through AI in an attempt to encourage the repurpose of clothing rather than purchasing them to an excessive degree. The system combines deep learning, virtual wardrobe management, and personalization for outfit suggestions to improve user experiences. OutfitX can simplify the outfit selection process, and coordination of fashion, and promote sustainability in the personal styling process. This paper will go further in the following sections with the methodologies, implementation, and evaluation of the system.

Evaluations conducted using the DeepFashion dataset revealed OutfitX exceeding baseline models such as Siamese Networks and collaborative filtering. The system integrates explainable artificial intelligence (XAI) to clarify the matching attributes in a recommendation, for example matching colors and harmonious patterns, which helps users trust the system more.

The dataset for OutfitX was sourced from both publicly available fashion databases and user-uploaded images. Two datasets have been used for this project. The primary dataset used is **DeepFashion2**, which contains 44,447 high-resolution images annotated with 13 clothing categories and 1,000 fine-grained attributes. Additionally, the **Polyvore dataset** provides curated outfit-level annotations, enabling the modeling of compatibility relationships for fine-tuning the Vision Transformer (ViT). Real-time contextual data, such as weather and event information, is sourced from APIs like **OpenWeatherMap** and **Google Calendar**. Preprocessing steps include image augmentation, segmentation using Mask R-CNN, and metadata tagging with ResNet-50. These datasets and preprocessing techniques ensure that the system is robust, accurate, and adaptable to diverse user needs. The datasets can be described as follows.

1.1. DATA SOURCING

Dataset 1

Source: UC Irvine Machine Learning Repository

Type: Synthetic

Characteristics: Dependent variables

Column types: Real, Object

Rows: 44,447 (No missing values)

Columns: 7

Variable Name	Role	Type
ID	Identification	Numerical
GENDER	Feature	Categorical
MASTER CATEGORY	Feature	Categorical
SUBCATEGORY	Feature	Categorical
ARTICLE TYPE	Feature	Categorical
BASE COLOUR	Feature	Categorical
SEASON	Feature	Categorical

- User images were categorized into three primary groups: upper body garments, lower body garments, and footwear. The dataset consisted of 44,447 images distributed as follows:
    1. Upper Body: **18,356 images**
    2. Lower Body: **16,230 images**
    3. Shoes: **9,861 images**
- It contains gender-based segmentation for Men, Women, Boys, Girls, and Unisex products.
- The masterCategory attribute represents the broad classification of items such as Apparel, Footwear, or Accessories.
- SubCategory refines the classification into types like Topwear, Bottomwear, or Bags.
- The articleType column specifies the detailed type of product, e.g., Shirts, Jeans, Handbags, or Shoes. Each item has a baseColour attribute representing its primary color (e.g., Black, Blue, Red).
- The dataset contains seasonal information like Summer, Winter, Fall, and Spring. It includes more than 200 unique article types.
- A large proportion of items are topwear garments like T-shirts, Shirts, and Sweatshirts. The dataset also features footwear categories like Casual Shoes, Flip Flops, and Heels.
- Accessories such as Watches, Bags, and Jewellery are well-represented. Seasonal distribution shows that Summer wear is the most dominant category.
- Color analysis reveals that Black, White, and Blue are the most common colors. The dataset contains both casual and formal wear.
- Gender-wise distribution indicates that Men’s apparel forms the largest portion. Footwear items are mainly available in the Summer and Winter seasons.
- The Personal Care category includes items such as Deodorants and Fragrances. Wallets and Bags are the dominant items in the Accessories category. The dataset includes a mix of traditional and modern clothing styles.

Dataset 2

This Polyvore dataset is a comprehensive collection of fashion outfits sourced from Polyvore.com, a platform where users curated and shared fashion ensembles. This dataset is important in fashion recommendation research, particularly in assessing outfit compatibility and item retrieval. This dataset contains:

- This dataset contains 21,889 outfits, divided into 17,316 for training, 1,497 for validation, and 3,076 for testing. Each outfit includes multiple fashion items curated by users.
- Contains images of individual fashion items within each outfit, facilitating visual analysis and modeling.
- Tagged each item with textual metadata, such as product descriptions and categories, enabling multi-modal analysis.
- Items are categorized into types like tops, bottoms, shoes, and accessories, aiding in structured analysis.
- Includes user-generated data, such as the number of likes per outfit, reflecting community preferences and item popularity.
- Provides information on which items are considered compatible within outfits, supporting compatibility modeling.
- Some items have seasonal labels (e.g., summer, winter), offering context for seasonal fashion trends. Details the primary colors of items, useful for color compatibility studies.

The columns may be described as follows.

- Image features
- Text features like Descriptive metadata and category labels, used in natural language processing (NLP).
- Categorical Features: Discrete labels such as item categories, colors, and seasons, used in classification tasks.

1.2. EXPLORATORY DATA ANALYSIS

This cardinal step was undertaken for both datasets, and the following insights were gathered. The pandas library was used for importing data and performing EDA.

	id	gender	masterCategory	subCategory	articleType	baseColour	season	usage
0	15970	Men	Apparel	Topwear	Shirts	Navy Blue	Fall	Casual
1	39386	Men	Apparel	Bottomwear	Jeans	Blue	Summer	Casual
3	21379	Men	Apparel	Bottomwear	Track Pants	Black	Fall	Casual
4	53759	Men	Apparel	Topwear	Tshirts	Grey	Summer	Casual
5	1855	Men	Apparel	Topwear	Tshirts	Grey	Summer	Casual
...	...	...	...	...	...	...	...	...
44417	12544	Women	Apparel	Topwear	Tshirts	Peach	Fall	Casual
44418	42234	Women	Apparel	Topwear	Tops	Blue	Summer	Casual
44419	17036	Men	Footwear	Footwear	Casual Shoes	White	Summer	Casual
44420	6461	Men	Footwear	Footwear	Flip Flops	Red	Summer	Casual
44421	18842	Men	Apparel	Topwear	Tshirts	Blue	Fall	Casual

Figure 1: Dataset Description

The unique base colors are present in the baseColour column of the dataset named styles. There are 43 unique base colors in the dataset. The colors include common shades like 'Blue', 'Black', 'Red', as well as more descriptive tones like 'Mushroom Brown', 'Sea Green', and 'Fluorescent Green'. The baseColour column is stored as an object type, which usually represents strings in pandas DataFrames.

```
len(styles.baseColour.unique()),styles.baseColour.unique()
```

```
(43,
 array(['Navy Blue', 'Blue', 'Black', 'Grey', 'Green', 'Purple', 'White',
        'Brown', 'Pink', 'Beige', 'Red', 'Khaki', 'Silver', 'Off White',
        'Yellow', 'Charcoal', 'Copper', 'Magenta', 'Orange', 'Lavender',
        'Sea Green', 'Cream', 'Peach', 'Gold', 'Tan', 'Olive', 'Burgundy',
        'Maroon', 'Grey Melange', 'Multi', 'Teal', 'Rust',
        'Turquoise Blue', 'Bronze', 'Mustard', 'Coffee Brown', 'Taupe',
        'Mauve', 'Mushroom Brown', 'Metallic', 'Nude', 'Lime Green',
        'Fluorescent Green'], dtype=object))
```

Figure 2: Unique colors extracted

This feature can be used for grouping items based on color, analyzing color preferences, and compatibility modeling between colors.

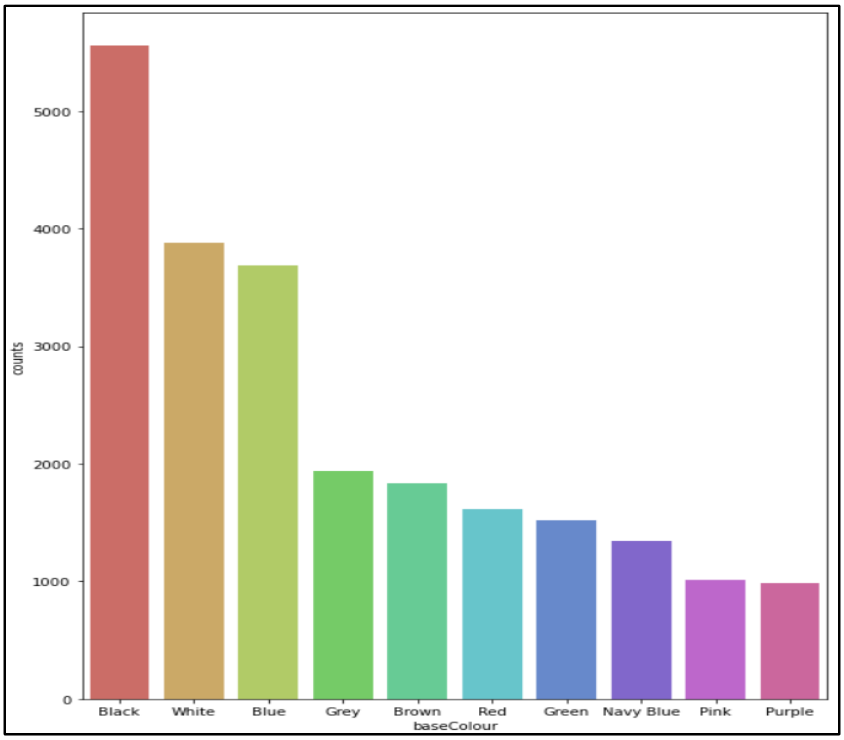


Figure 3: Bar Chart of colors extracted

Black is the most frequently occurring color, followed by White and Blue. The counts decrease gradually, with Purple having the lowest count among the top 10. The chart provides a percentage-based visual representation of the color distribution. Black occupies the largest percentage (20.4%), followed by White (14.2%) and Blue (13.5%). The smaller segments represent less frequent colors.

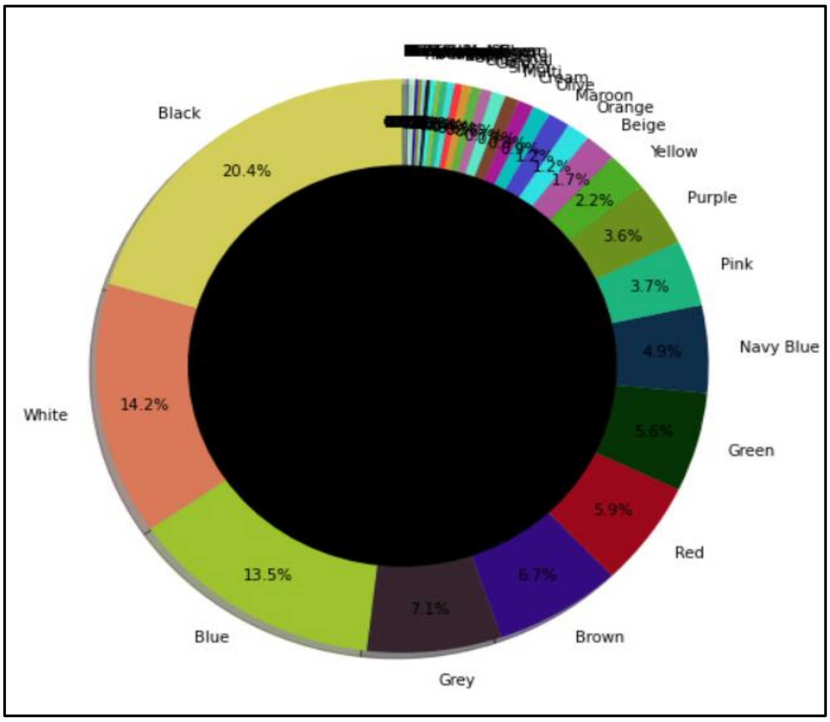


Figure 4: Color Distribution Pie Chart

This bar chart shows the Distribution of Clothing Categories, with the category_name on the y-axis and count on the x-axis. Short sleeve tops dominate the dataset, with a significantly higher count compared to other categories. This indicates that casual wear like tops is more common in the wardrobe, potentially due to its versatility across seasons.

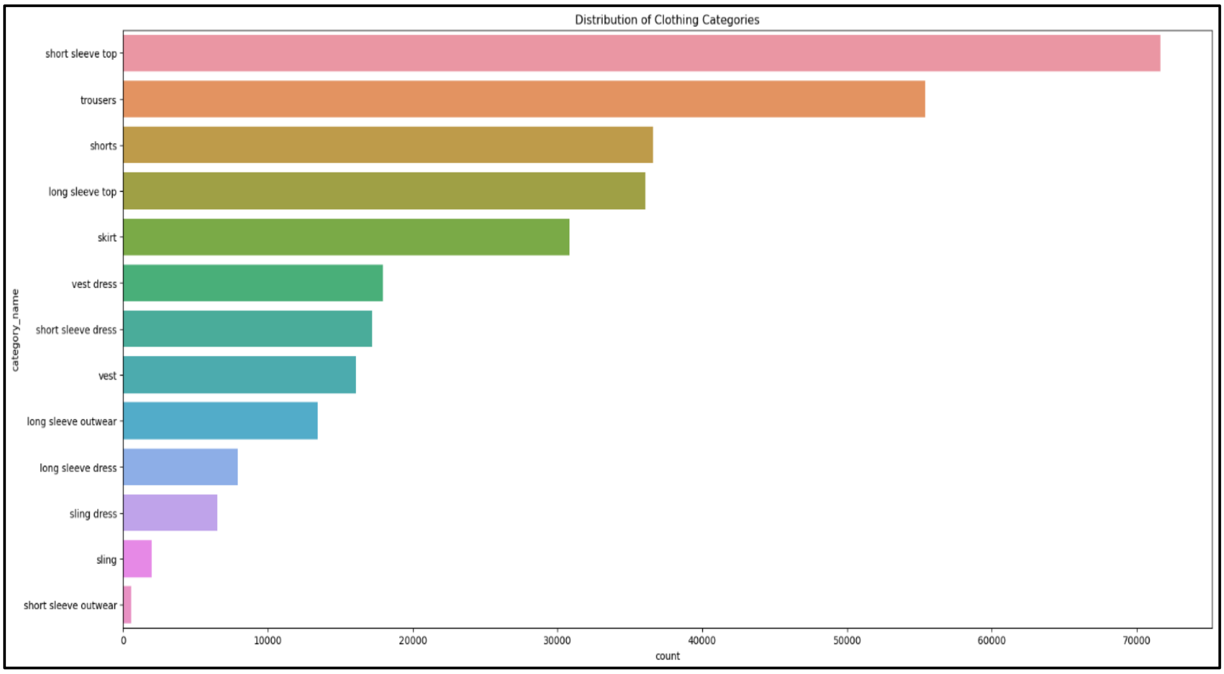


Figure 5: Bar Chart showing distribution of clothing categories

The histogram shows a right-skewed distribution, indicating that most bounding boxes are smaller in size. Smaller bounding boxes could represent accessories or smaller apparel items like belts, shoes, or handbags. The peak between 5,000 to 10,000 area pixels suggests that most images contain small to medium-sized objects.

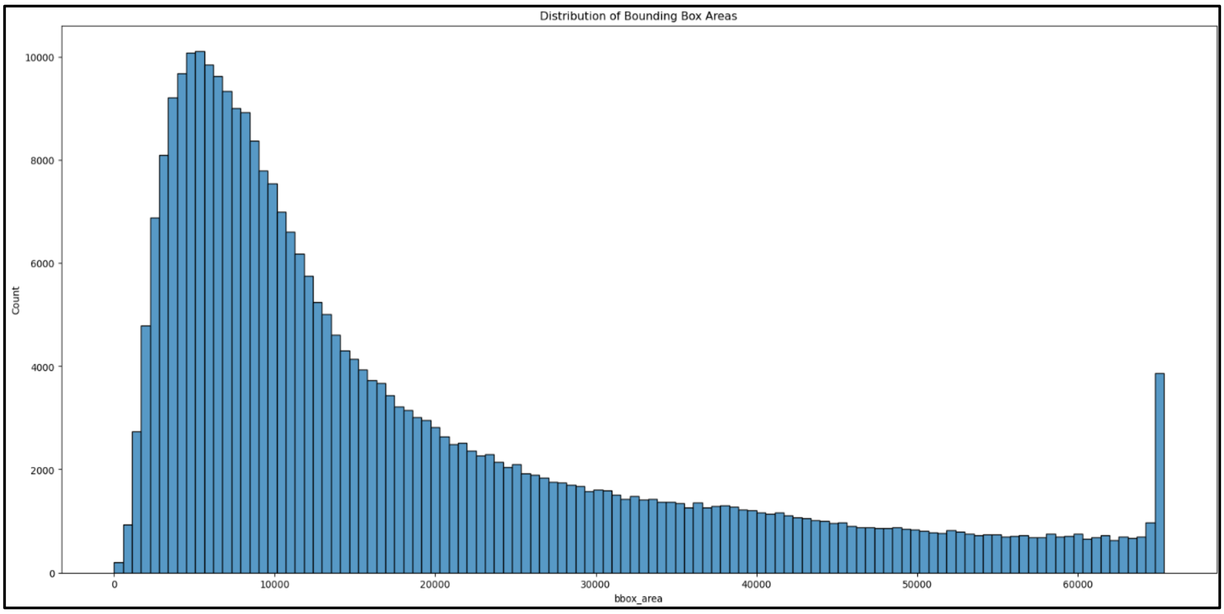


Figure 6: Histogram showing bounding box area distribution

The viewpoint is the dominant perspective across all categories, especially for short sleeve tops, trousers, and skirts. Side/back viewpoints are significantly fewer in number, which may affect model performance for these views. The "no wear" category appears consistently but in smaller quantities.

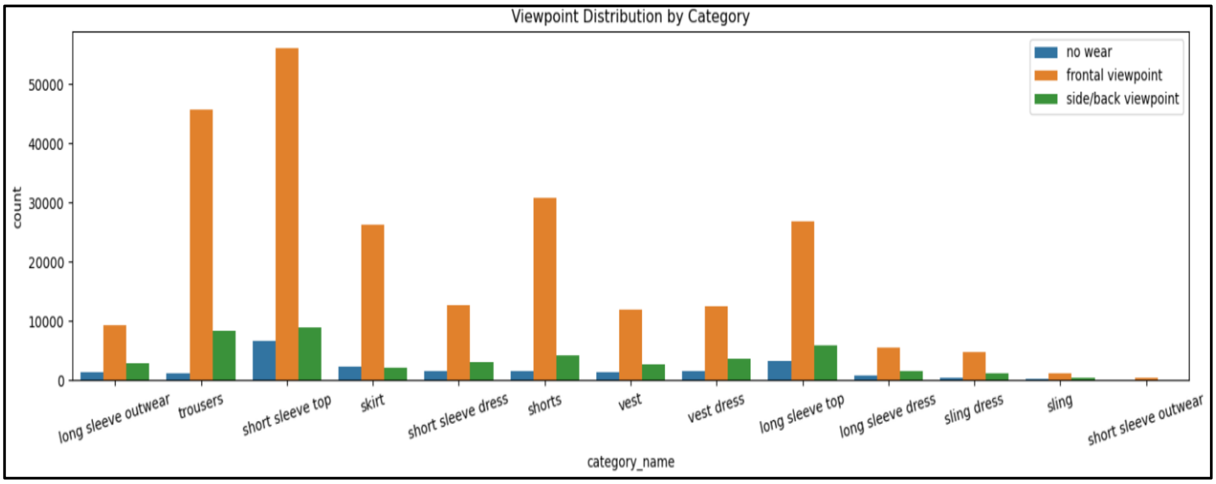


Figure 7: Stacked Bar Chart showing category viewpoint distribution

Most bounding boxes have smaller areas, indicating that most detected objects are relatively small. There is a long tail of larger bounding boxes, showing fewer large objects. A spike at the right suggests that a certain size threshold is frequently occurring, possibly representing full-body images or large product displays.

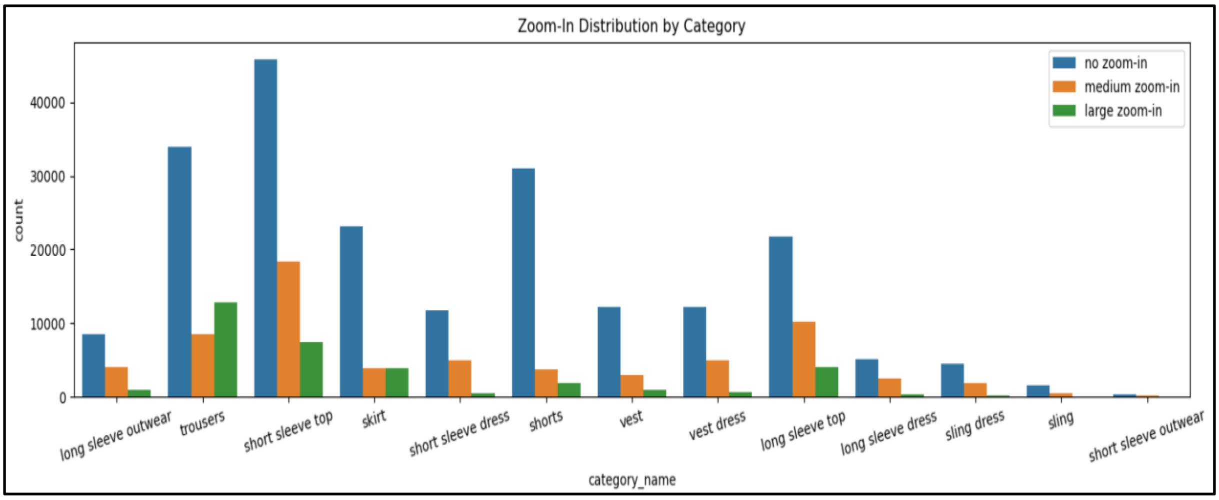


Figure 8: Stacked Bar chart bounding box area distribution

Most images have no zoom-in across all categories. Medium zoom-in is the second most common, especially in categories like skirts and tops. Large zoom-in is rare, indicating that close-up shots are less common.

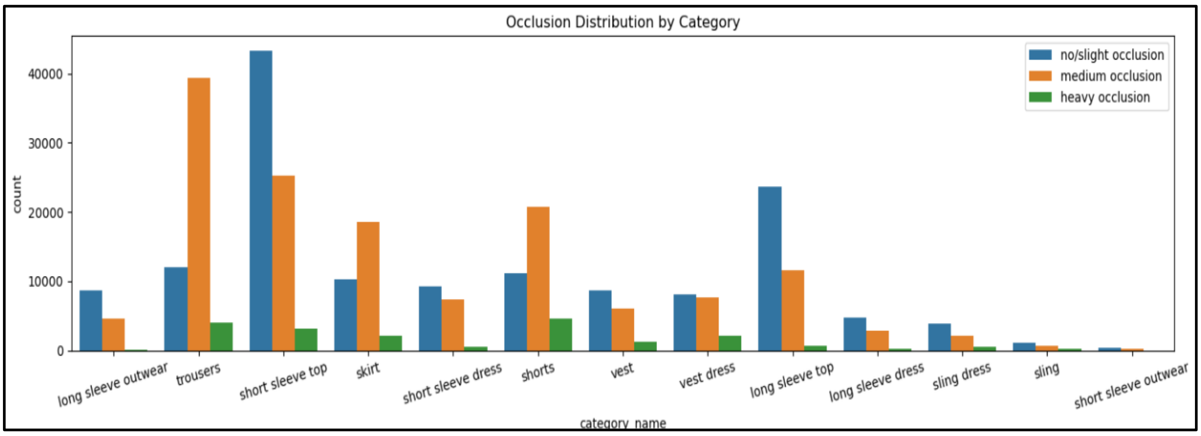


Figure 9: Occlusion distribution by category

Most images show no/slight occlusion, which is ideal for model training. Medium occlusion is common, especially in trousers, skirts, and dresses. Heavy occlusion is rare, making it a challenging subset for the model.

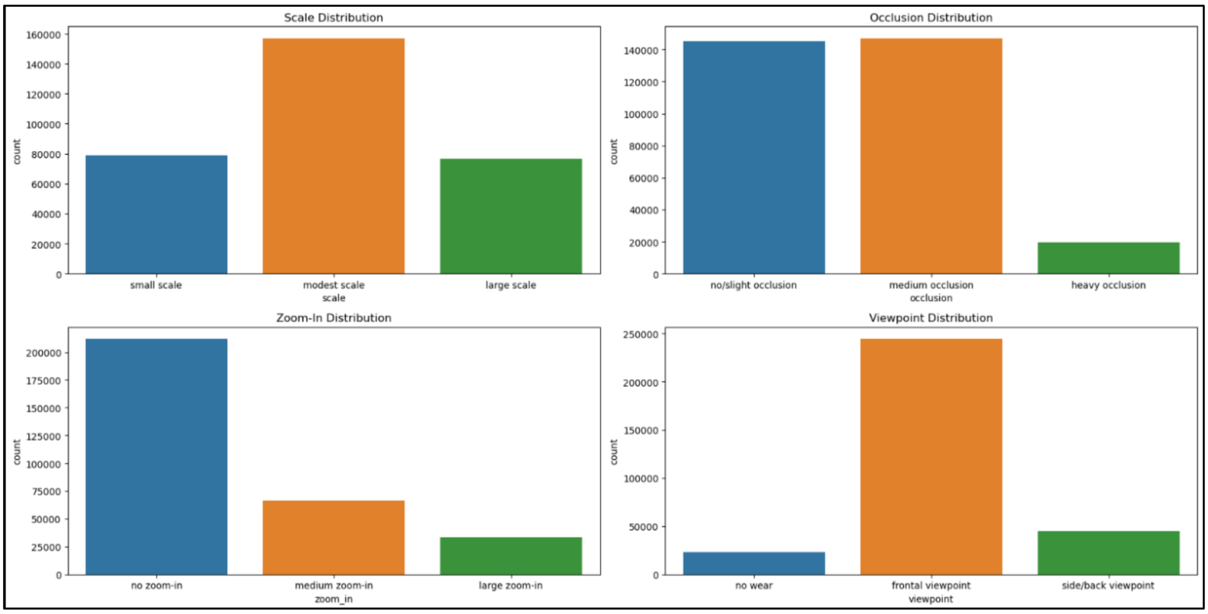


Figure 10: Bar chart distributions

The dataset might be biased towards modest-sized objects, which could impact model performance on small or large objects. The dataset is imbalanced in terms of occlusion, which means the model might struggle to detect objects under heavy occlusion. The dataset could have more full-frame objects rather than close-ups, which might affect the model’s ability to detect fine details. The dataset is biased towards front-facing objects, which could limit the model’s performance on side or back views. A data augmentation strategy or weighted loss function could help mitigate this imbalance. The model might perform better on modest scale, frontal viewpoint, and no occlusion images.

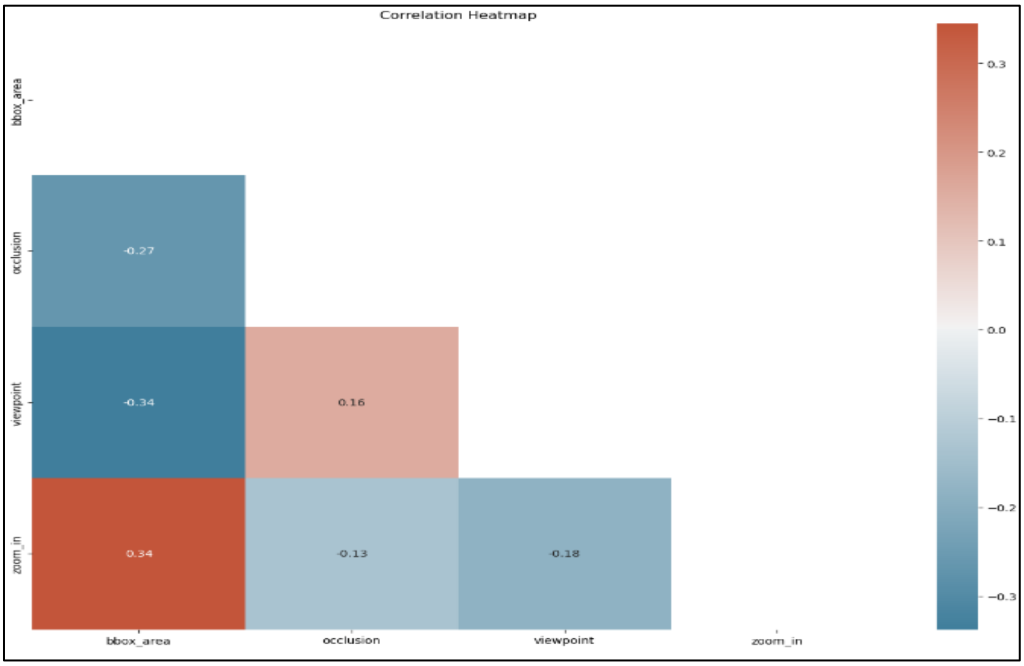


Figure 11: Heat Map to check for correlation

The bbox_area (Bounding Box Area) shows a positive correlation (0.34) with zoom_in, indicating that larger bounding box areas are often associated with zoomed-in images. There is a negative correlation (-0.34) between bbox_area and viewpoint, implying that certain viewpoints tend to have smaller bounding box areas. Occlusion and bbox_area have a weak negative correlation (-0.27), suggesting that more occluded objects are often smaller in area. Viewpoint and occlusion show a small positive correlation (0.16), indicating that some viewpoints might slightly affect the level of occlusion.

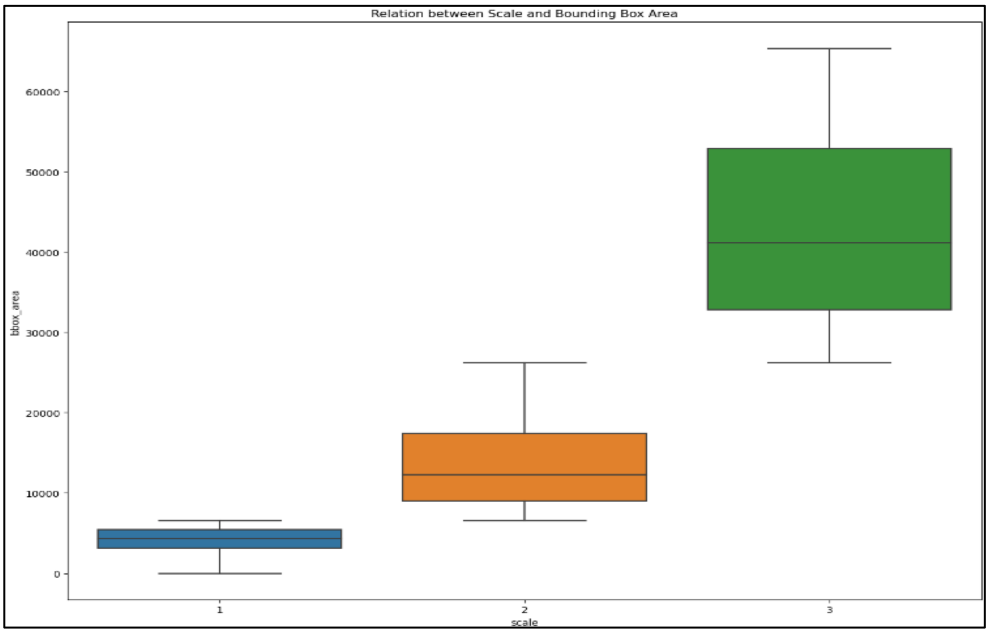


Figure 12: Box plot between Scale and Bounding Box Area

The bounding box area increases significantly with the scale. Scale 1 represents smaller objects with lower area values. Scale 2 shows moderate object sizes with higher bounding box areas. Scale 3 has the largest bounding box areas, showing that larger objects have higher area measurements. The object size and occlusion patterns are influenced by scale, zoom level, and viewpoints.

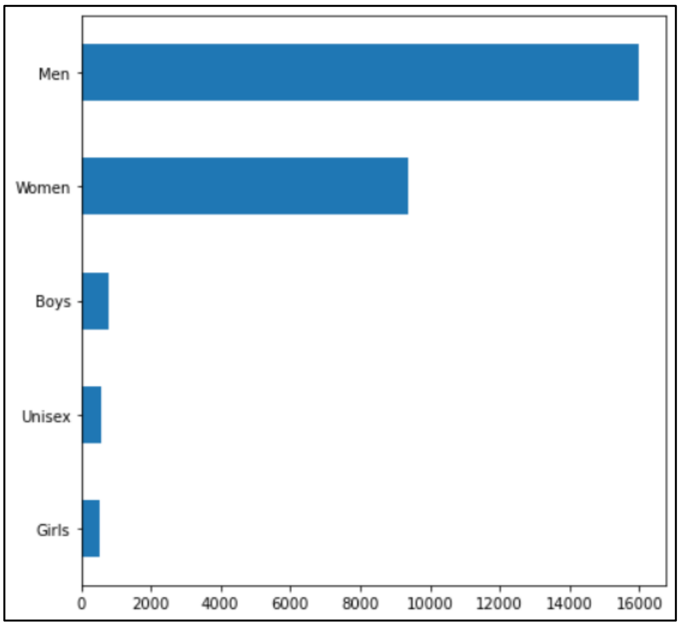


Figure 13: Gender Categorization

The dataset is heavily skewed towards men's outfits, indicating a possible bias in data collection. If this dataset is used for classification tasks (e.g., recommending based on gender), the model might struggle with class imbalance, especially for Girls and Unisex. The dataset could suffer from sparsity, making it difficult to learn patterns from these categories. This imbalance might affect the model's performance, especially when recommending outfits for women or children. The small proportion of Unisex outfits indicates that gender-neutral fashion is underrepresented. The significant gap between adult categories (Men, Women) and child categories (Boys, Girls) highlights that the dataset might not be diverse enough to cater to family-oriented or kids' fashion brands. To improve recommendation accuracy, especially for underrepresented categories like Unisex, Girls, and Boys, data augmentation, by increasing the size of underrepresented categories through image transformations or synthetic data generation, weight adjustments, assigning higher weights to underrepresented categories during model training, could help balance the dataset. The model could be fine-tuned to perform better on children's clothing with smaller datasets.

We see that 200 components explain approximately 95% variance and 80 components explain 90% variance. As the number of components increase, the explained variance rises to 100%, which makes sense because it is original matrix with 784 dimensions. Each color represents a label.

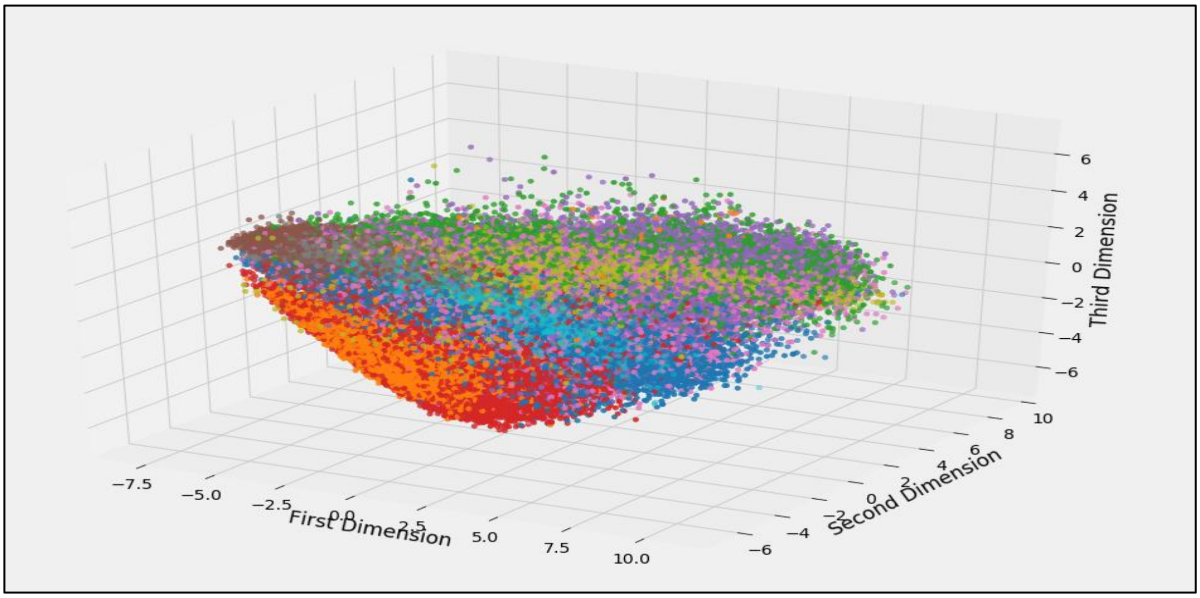


Figure 14: Two components can separate different categories

This t-SNE plot visualizes high-dimensional data in two dimensions. Each color corresponds to a different clothing category such as T-shirt, Trouser, Pullover, Dress, etc. Categories like Trousers and Bags seem well-separated, indicating the model can distinguish them effectively. Categories like T-shirts, Pullovers, and Coats show overlaps, suggesting these items might have more visual similarities, making them harder to classify.

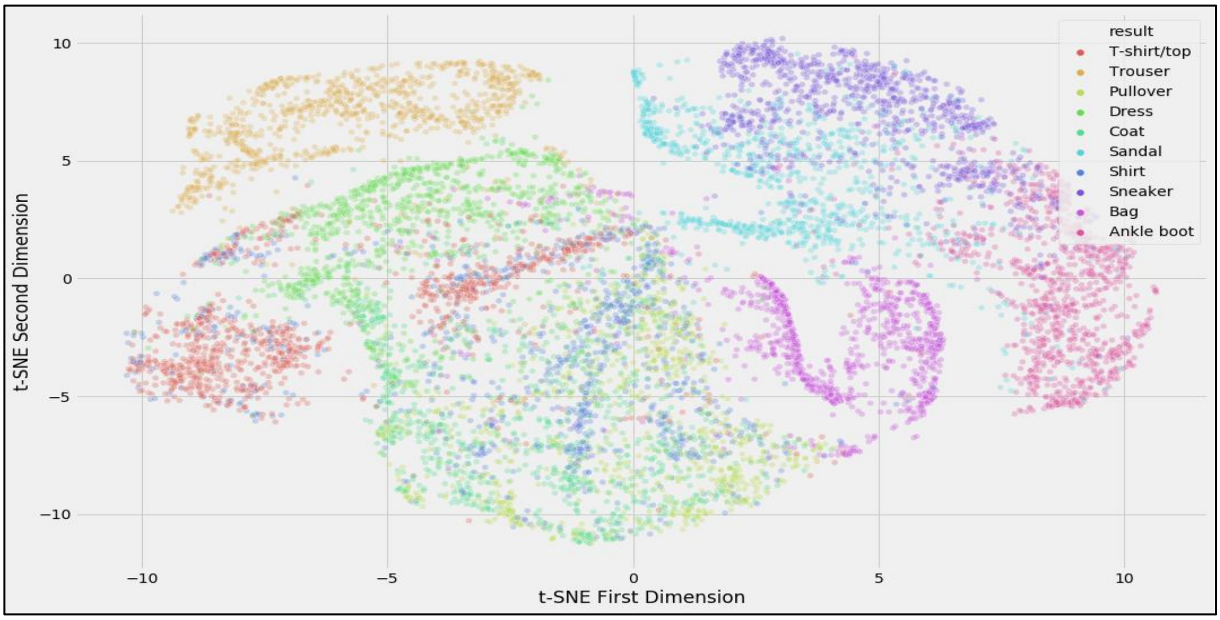


Figure 15: Latent Features

2. REVIEW OF LITERATURE

This section looks at different advancements in using deep learning in fashion technology. Deep learning is a method that can learn from data and make predictions. The paper also talks about how much these methods cost and how well they are. We discuss the challenges of using these methods and suggest future advancements.

Convolutional neural networks (CNNs) are used for personalization, and Pinterest’s Visual Search retrieves visually similar items from the database based on embedding generated by CNNs from the user-uploaded images. This is way greater than nothing, but it doesn’t provide a means to model outfit-level compatibility or user-specific preferences. An example of this is people searching for striped shirts when they only see visual matches, but have no matches (for example, getting them dressed tailored with more tailored pants)[1], [5].

The author has developed a new deep reinforcement learning (DRL) method called SustainFashion to create sustainable outfit recommendations. The system implements LCA data through its reward mechanism to select clothing components made from organic cotton and recycled polyester and items with minimal carbon emissions. The system suggests a linen shirt instead of a polyester shirt since linen generates less environmental impact. The evaluation of the framework took place on a specially designed dataset of 10,000 outfits that included sustainability annotations. Through SustainFashion the product recommendations lowered carbon emissions by 23% versus conventional methods while users kept 85% satisfaction with the results. The multi-objective reward function stands out as a core innovation since it links sustainability metrics to user preferences for ideal recommendations between style and practicality. Manual curation of LCA data represents a challenge for scalability because such information tends to be classified as proprietary or contains incomplete sections. The development of automated sustainability metric estimation technology from product descriptions and images would help expand system accessibility in later stages of research [3].

In early methods such as SIFT-based feature matching, the clothing items were identified using descriptors such as edges and corners. Nevertheless, as shown above, these are computationally efficient methods and resulted in poor performance on intra-class variability (e.g., separating a floral sundress from a floral blouse) and did not make use of semantic attributes of articles (occasion or season). In the context of Siamese networks, their two networks are trained to minimize the distance between compatible items (shirts and pants) and maximize the distance between incompatible items. However, they still treated the task as a binary one without imposing a hierarchy between items (e.g., how the texture of a blazer is connected to recommended shoes)[6], [7].

Graph-based recommendation systems model the outfits as graphs (items represented as nodes, compatibility of nodes with each other represented as edges), and, perhaps more importantly, representation of outfits using word embeddings. More specifically, these systems often rely on having a set of friendly colors complement rather than learning pairwise compatibility dynamically from user behavior. OutfitX goes further in this paradigm and accordingly extends it to incorporate users' specific metadata, e.g., past preferences and local weather, into a GNN that can make adaptive recommendations based on preferences determined by the dynamic. [8], [9].

Deep learning is being implemented in fashion analytics, especially in attribute prediction and generative design. ModaNet dataset of 55,000 high-resolution images annotated for 13 categories and 37 attributes (e.g. "neckline" and "dress") of 13 classes is an example. Models trained on ModaNet result in 85% mAP for attribute prediction but struggle in the rare or exceptional categories (e.g., "bohemian style"). The generation of the entire outfit by generative models such as Outfit-GAN is done using c-GANs. These models suffer from a mode collapse, usually providing very boring or impossible-to-achieve combinations (like putting winter coats on with sandals). Also, they are unexplainable to users, and thus, they cannot trust them nor can they refine the suggestions further. Multi-modal fusion between visual features and textual descriptions like product reviews is required to make the recommendation accurate. For example, items retrieved in context are offered by BERT-based models based on queries such as 'office appropriate dresses'. While these systems are effective, they are limited for large labeled datasets that limit their market only to niche markets or personalized wardrobes. This is the trade-off for the drawback of this method, which OutfitX solves by using self-supervised learning on unlabeled user-uploaded images and minimizing the dependency on curated datasets.[10].

The research highlights FusionNet as an architectural integration of visual features extracted from ResNet-50 and textual descriptions encoded with BERT embeddings with the purpose of outfit recommendations. FusionNet uses cross-modal attention to properly link the visual segments with textual components for context-aware suggestion generation. The model delivers advice for a beach outing by examining both the visual design of a bohemian dress and the textual elements of the question. The researchers evaluated FusionNet using the Amazon Fashion dataset consisting of 200,000 products that contain both image data and product descriptions. FusionNet reached 87.4% recommendation accuracy, making it 15% better than baselines that used only single-modal data. The model experienced difficulties when processing product descriptions that were not in English, thus indicating a requirement for multilingual adaptation. The pairing requirement for FusionNet's data source makes it less effective for working with datasets with insufficient or noisy metadata. Future research should aim to establish methods that use either no labels or minimal supervision to combat labeled data requirements.[12].

This paper introduces the application of Transformer models that were used in processing sequential data, of which the most popular examples include outfit sequence, with the use of self-attention mechanisms. For example, the Outfit Transformer encodes outfit items as tokens and independently predicts the compatibility scores of outfit features based on inter-item relationships. When compared with CNNs and RNNs, it outperforms these models in capturing long-range dependencies (i.e., dependence between a handbag and a shoe choice), while at the same time being computationally expensive and thus not applicable to real-time systems.[13].

In this paper, the model achieves fashion compatibility through a method called Knowledge Graphs. KGs consider clothing items, attributes, and relationships as a structured graph to be semantically reasoned such that "a wool blazer is good for winter events." The combination of KGs and RL has been used to optimize user feedback-based outfit recommendations. One such use case is that KG-RL was able to achieve 88% precision on the Fashion-Gen dataset iteratively, getting increasingly refined recommendations using user interactions. These methods handle cold start scenarios poorly as there is less data collection regarding the users.[14].

In this manuscript, the author has used StyleFlow, a framework based on normalizing flows to separate and modify style elements within fashion imagery. The generative models called normalizing flows teach inverted transformations between complicated data distributions and straightforward latent distributions, which allow exact attribute control. StyleFlow transforms clothing item latent spaces through hierarchical flows that enable users to change sleeve length, neckline design, and color choice without compromising garment structure. Users can transform extensive-sleeved shirts into short-sleeved garments, maintaining the original design structure. The evaluation of the framework occurred on the DeepFashion-C dataset using its collection of 52,000 annotated images at high resolution alongside 1,000 detailed attributes. The 89.2% accuracy rating of StyleFlow exceeded the 12% increase over comparable GAN-based baselines such as StyleGAN.[15].

This paper shows that Meta-learning has also been developed as a solution to personalized recommendations. Meta-learning models do this by implementing a few shot-learning techniques to learn quickly under similar users’ conditions with very little data. This makes the system faster and rapid than the usual learning rate. Meta Fashion fine-tunes the pre-trained CNNs on the small user-specific dataset and achieves 82% accuracy in predicting user preference. However, these models are very good, but hyperparameters are very tuned, and these are very biased towards the data set.[16]

3. PRESENT WORK

3.1 PROBLEM FORMULATION

Traditional fashion recommendation systems primarily rely on user ratings, purchase history, or textual descriptions, which often fail to capture the visual aspects of outfits. However, fashion is a highly visual domain where users tend to make purchasing decisions based on the appearance of clothing items. This gap creates a need for advanced recommendation systems that integrate both visual and contextual information to provide more accurate and personalized outfit suggestions. The lack of such systems highlights the problem of inadequate utilization of visual data in generating outfit recommendations, leading to unsatisfactory results for users seeking style inspiration.

The main challenge in developing an outfit recommendation system is effectively analyzing clothing images and identifying key features such as color, texture, and category. Additionally, recommending appropriate combinations of upper wear, lower wear, footwear, and accessories requires understanding how different fashion items complement each other. This study aims to formulate an AI-based Outfit Recommendation System (OutfitX) that leverages computer vision and machine learning techniques to address these challenges. The system allows users to upload images of their clothing items to create a virtual wardrobe, where each item is automatically categorized into predefined fashion categories. By extracting visual features from the uploaded images, the system will generate personalized outfit suggestions that align with user preferences.

Traditional recommendation systems focus on popularity over personalization, which is why users end up having no wish-related suggestions. For instance, collaborative filtering cannot handle the “cold start” problem for new users, and the content such as Siamese networks does not deal with pairwise compatibility while never capturing the balanced nonpairwise aspects of holistic cohesion fulfilled context. The users are not aware of why certain items are recommended; these systems are not transparent. Taking fashion as an example, this gap is critical in that subjective preferences, cultural trends, and other unknowns play a much greater role in decisions. Traditional fashion apps depend on predefined templates, to analyze user data in user wardrobes to ensure personalized outfit recommendations. To improve this, we recommend OutfitX, a virtual wardrobe system that works with deep learning and image classification to support users when selecting outfits based on uploaded clothing images.

3.2 OBJECTIVE OF STUDY

Traditional recommendation systems focus on popularity over personalization, which is why users end up having no wish-related suggestions. For instance, collaborative filtering cannot handle the "cold start" problem for new users, and the content such as Siamese networks does not deal with pairwise compatibility while never capturing the balanced nonpairwise aspects of holistic cohesion fulfilled context. The users are not aware of why certain items are recommended; these systems are not transparent. Taking fashion as an example, this gap is critical in that subjective preferences, cultural trends, and other unknowns play a much greater role in decisions. The objective of this study was an AI-based Outfit Recommendation System (OutfitX) that leverages computer vision and machine learning techniques to accurately suggest outfits based on user-uploaded clothing images.

- Enhance fashion coordination by utilizing color theory, style compatibility, and seasonal factors to generate well-matched outfit recommendations.
- Implement a smart digital wardrobe that allows users to store, categorize, and manage their clothing items digitally for easier selection.
- Extend OutfitX's capabilities to virtual try-ons, online shopping integration, and AI-driven trend analysis.

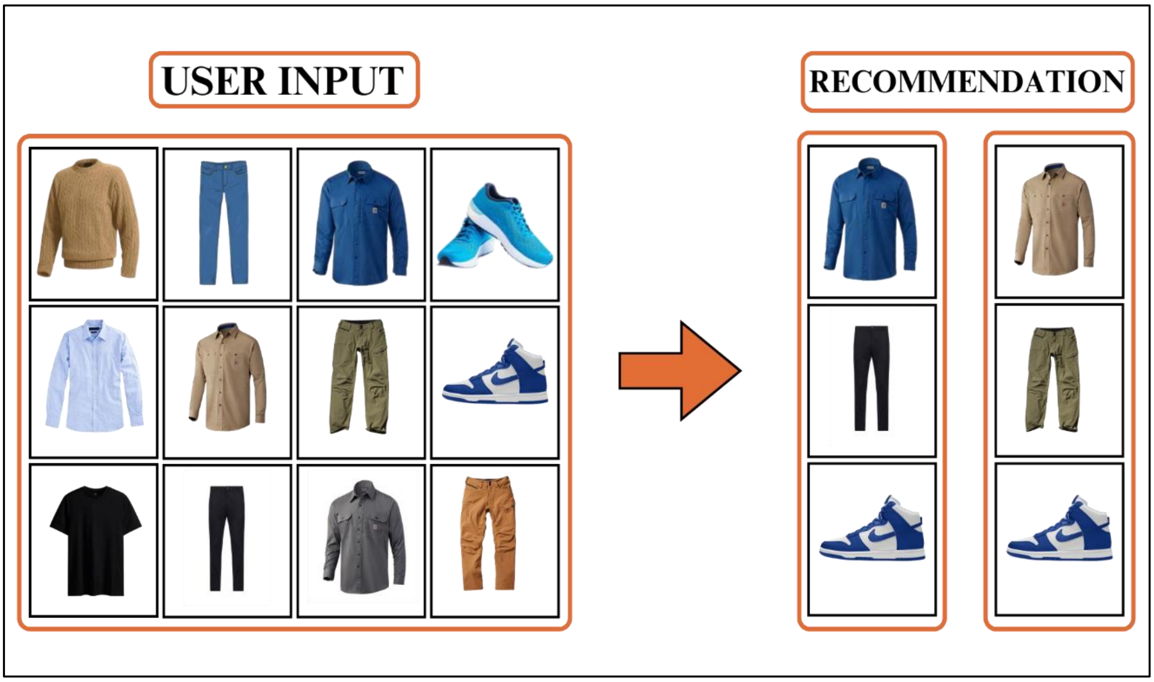


Figure 16: Functioning of OutfitX Recommendation System

3.3 RESEARCH METHODOLOGY

In this section, we discussed the research methodology for a deep-learning model that can help with customized outfit recommendations by processing images. We have used major steps to use the images and train the model. The steps involved (A) Data Collection and Preprocessing, (B) Feature Extraction and Digital Wardrobe, (C) Compatibility Modeling with Graph Neural Networks (GNN), (D) Contextual Metadata Fusion, and (E) Explainability (XAI).

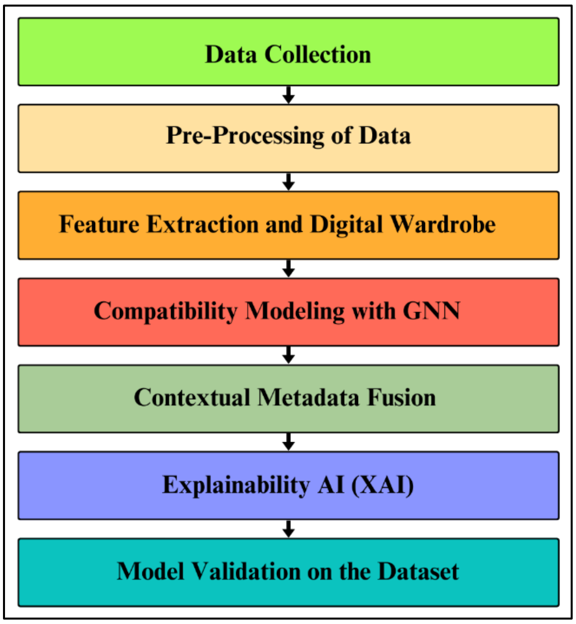


Figure 17: Detailed methodology of OutfitX System Architecture

The system depends on a combined CNN-Transformer architecture that analyzes user UPLOADED CLOTHING IMAGES to obtain both primary visual characteristics and advanced semantic aspects. A digital wardrobe utilizes the extracted features to store user items as nodes within a model where edges represent compatibility connections between them. The algorithm implements Graph Attention Networks (GATs) to understand how different clothing items match through multi-head attention processes while learning dynamic pairwise relations between them. Through this weight adjustment process, the system performs better recommendations by incorporating actual-world factors, including seasonal changes, user preferences, and event types for creating more personalized and context-aware suggestions. The Explainable AI (XAI) combines Grad-CAM to extract visual features that affect outfit suggestions. The model is evaluated through Precision and Cumulative Gain (NDCG) metrics. The training process is fine-tuned using the Adam optimizer with hyperparameters, a learning rate of 0.001, and a 64-batch size.

4. IMPLEMENTATION AND VALIDATION

4.1 INTRODUCTION TO CNN

Convolutional Neural Networks (CNNs) are the cornerstone of modern computer vision, adept at processing visual data with remarkable precision and efficiency. Operating on the principle of hierarchical feature learning, CNNs autonomously extract intricate patterns and features directly from raw pixel data. Through layers of convolutional operations, these networks detect low-level features like edges and textures, gradually synthesizing them into higher-level representations of shapes and objects. Interspersed pooling layers help condense information, while activation functions inject non-linearity, enabling CNNs to capture complex relationships within the data. With fully connected layers at the network's culmination, CNNs interpret learned features to perform tasks such as image classification, object detection, and more. Trained through backpropagation and optimization techniques, CNNs continually refine their parameters to minimize error and enhance accuracy. From medical imaging to autonomous vehicles, CNNs have revolutionized diverse fields by endowing machines with the ability to comprehend and analyze visual information, heralding a new era of artificial intelligence.

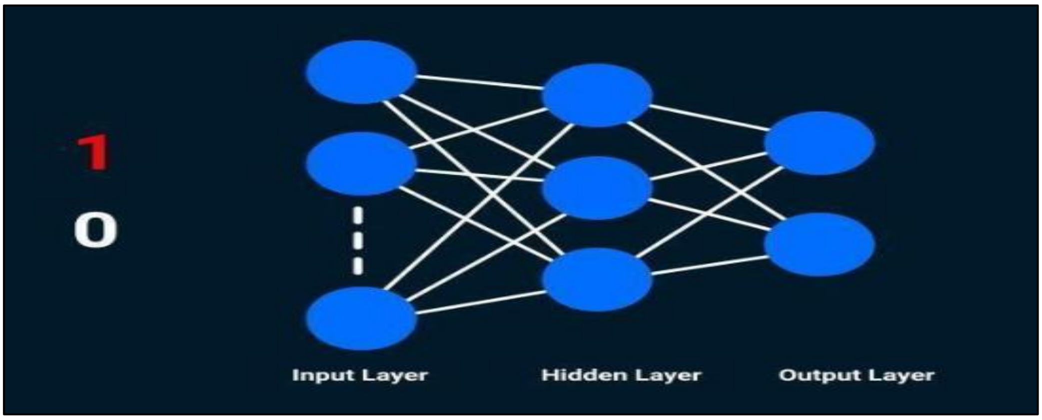


Figure 18: CNN Architecture

Convolutional Neural Networks (CNNs) are a pivotal component of contemporary computer vision systems, revolutionizing how machines perceive and interpret visual data. By leveraging hierarchical feature learning, CNNs autonomously extract intricate patterns and features directly from raw pixel data. Through successive layers of convolutional operations, these networks progressively detect low-level features such as edges and textures, gradually synthesizing them into higher-level representations of shapes and objects. Interspersed pooling layers facilitate information condensation, while activation functions introduce nonlinearity, enabling CNNs to capture complex relationships within the data. Fully connected layers at the network's apex interpret learned features to perform tasks like image classification and object detection. Trained via backpropagation and optimization, CNNs continuously refine their parameters to minimize error and enhance accuracy. From medical imaging to autonomous vehicles, CNNs have catalysed advancements across diverse domains by endowing machines with the ability to comprehend and analyze visual information, heralding a new era of artificial intelligence.

4.2 BASIC PRINCIPLE OF CNN

Convolutional layers are fundamental components of Convolutional Neural Networks (CNNs). These layers apply learnable filters or kernels to input data, enabling the network to extract essential features from the input images. Each filter acts as a feature detector, scanning the input image for patterns such as edges, textures, or shapes. By convolving the filters with the input data, convolutional layers can capture meaningful spatial hierarchies and representations crucial for subsequent processing tasks. The convolution operation involves sliding a filter matrix (kernel) over the input image and computing the dot product at each spatial location. Mathematically, this can be expressed as follows:

$$\boxed{(f*g)(x,y)=\sum m\sum nf(m,n)\cdot g(x-m,y-n)(f*g)(x,y)=\sum m\sum n\ f(m,n)\cdot g(x-m,y-n)}$$

Here, $(f*g)(x,y)(f*g)(x,y)$ represents the output value at spatial location (x,y), obtained by convolving filter g with input data f. The summation is performed over the filter dimensions m and n, covering all overlapping regions between the filter and input image. To visualize the convolution operation, imagine sliding a small matrix (filter) over an image, starting from the top-left corner. At each position, the filter overlaps with a patch of the input image. The corresponding elements of the filter and the input patch are multiplied elementwise, and the results are summed to produce a single value in the output feature map. By sliding the filter across the entire image, the convolutional layer generates feature maps that capture spatial patterns and structures present in the input data. These feature maps serve as representations of learned features that are subsequently used for tasks such as classification, detection, or segmentation. Convolutional layers play a crucial role in CNNs by extracting informative features from raw input data through the convolution operation. This process enables CNNs to effectively process and analyze visual information in computer vision applications.

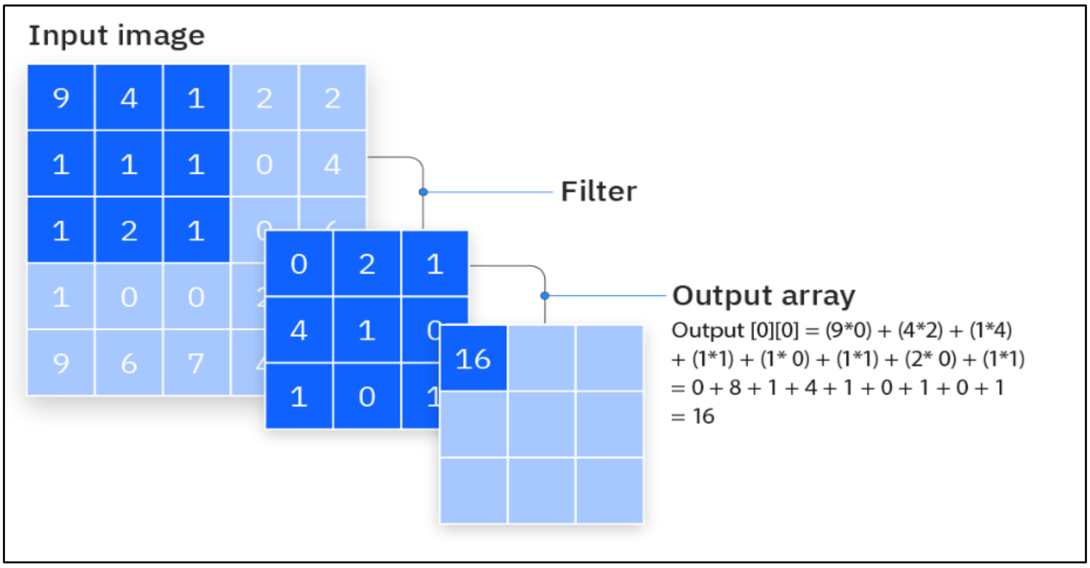


Figure 19: Filtering on Input Features

Pooling layers in Convolutional Neural Networks (CNNs) are designed to reduce the spatial dimensions of feature maps generated by convolutional layers while retaining important information. By down sampling the feature maps, pooling layers help in reducing computational complexity, controlling overfitting, and extracting robust features invariant to small spatial translations. There are several common pooling methods used in CNNs.

In max pooling, each region of the feature map is divided into smaller sections (usually non-overlapping), and the maximum pixel value within each region is retained. Max pooling effectively preserves the most dominant features present in each region. In average pooling, each region of the feature map is partitioned similarly to max pooling, but instead of retaining the maximum pixel value, the average pixel value within each region is calculated and retained. Average pooling provides a smoothed representation of the feature map.

Activation functions are critical components of neural networks, including Convolutional Neural Networks (CNNs). They introduce non-linearity into the network, enabling it to model complex patterns and relationships within the data. Without activation functions, neural networks would merely be linear transformations of the input data, severely limiting their expressive power and learning capacity. Activation functions are applied to the output of each neuron in a network, allowing it to introduce non-linearities in the network's decision boundaries and enabling the network to learn and represent intricate features and relationships in the data. There are various types of activation functions used in neural networks, each with its characteristics and applications.

ReLU is one of the most widely used activation functions in deep learning. It is defined as $f(x)=max(0,x)$, meaning it returns zero for any negative input and passes positive input values unchanged. ReLU is preferred due to its simplicity, computational efficiency, and ability to alleviate the vanishing gradient problem. The sigmoid activation function, also known as the logistic function, maps input values to the range $(0,1)$. It is often used in binary classification tasks, where the output needs to be interpreted as probabilities. Tanh is similar to the sigmoid function, but maps input values to the range $(-1,1)$. It is commonly used in hidden layers of neural networks, allowing for better gradient propagation compared to sigmoid. SoftMax is primarily used in the output layer of classification networks, where it normalizes the output values into a probability distribution over multiple classes, ensuring that the sum of the probabilities equals one.

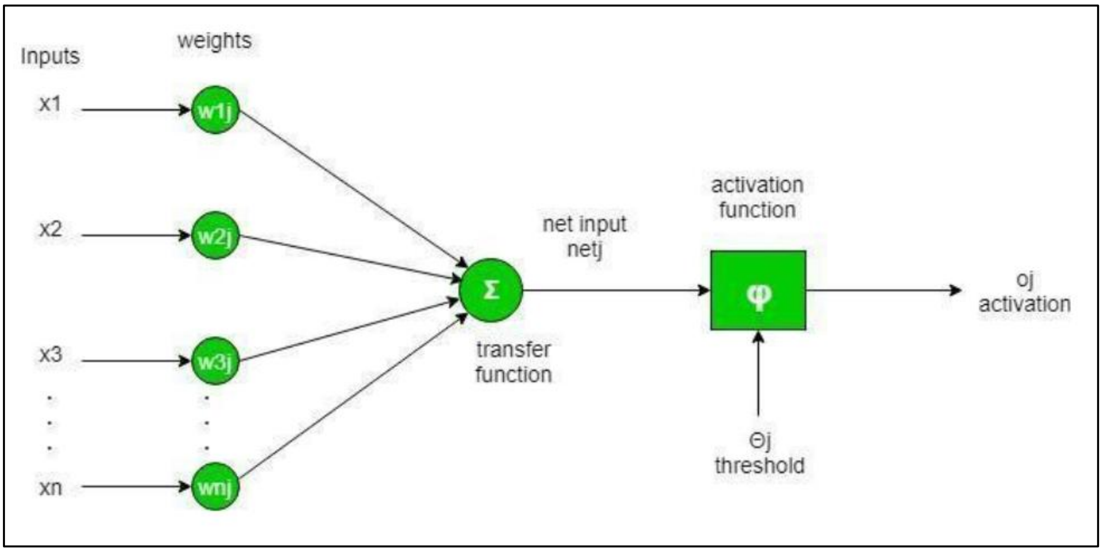


Figure 20: Activation Function

ReLU introduces a hinge-like non-linearity at zero, allowing the network to selectively activate neurons based on the positive input values. Sigmoid and tanh functions introduce S-shaped curves, squashing input values into specific ranges. SoftMax transforms the network's raw outputs into a probability distribution, with higher values indicating higher probabilities for the corresponding classes. Activation functions play a crucial role in determining the capacity and behavior of neural networks. By introducing non-linearities, they enable networks to learn complex patterns and relationships, making them indispensable components.

4.3 DATA AUGMENTATION

OutfitX uses the DeepFashion2 dataset, containing 44,447 high-definition pictures with labeling into 13 different categories while also attaching 1,000 fashion attributes with their examples, including "tops," "dresses," "striped," and "v-neck." The model uses 37,413 images to build its simulated personalized wardrobe because these images include varied styles and multiple color options for various occasions and seasons.

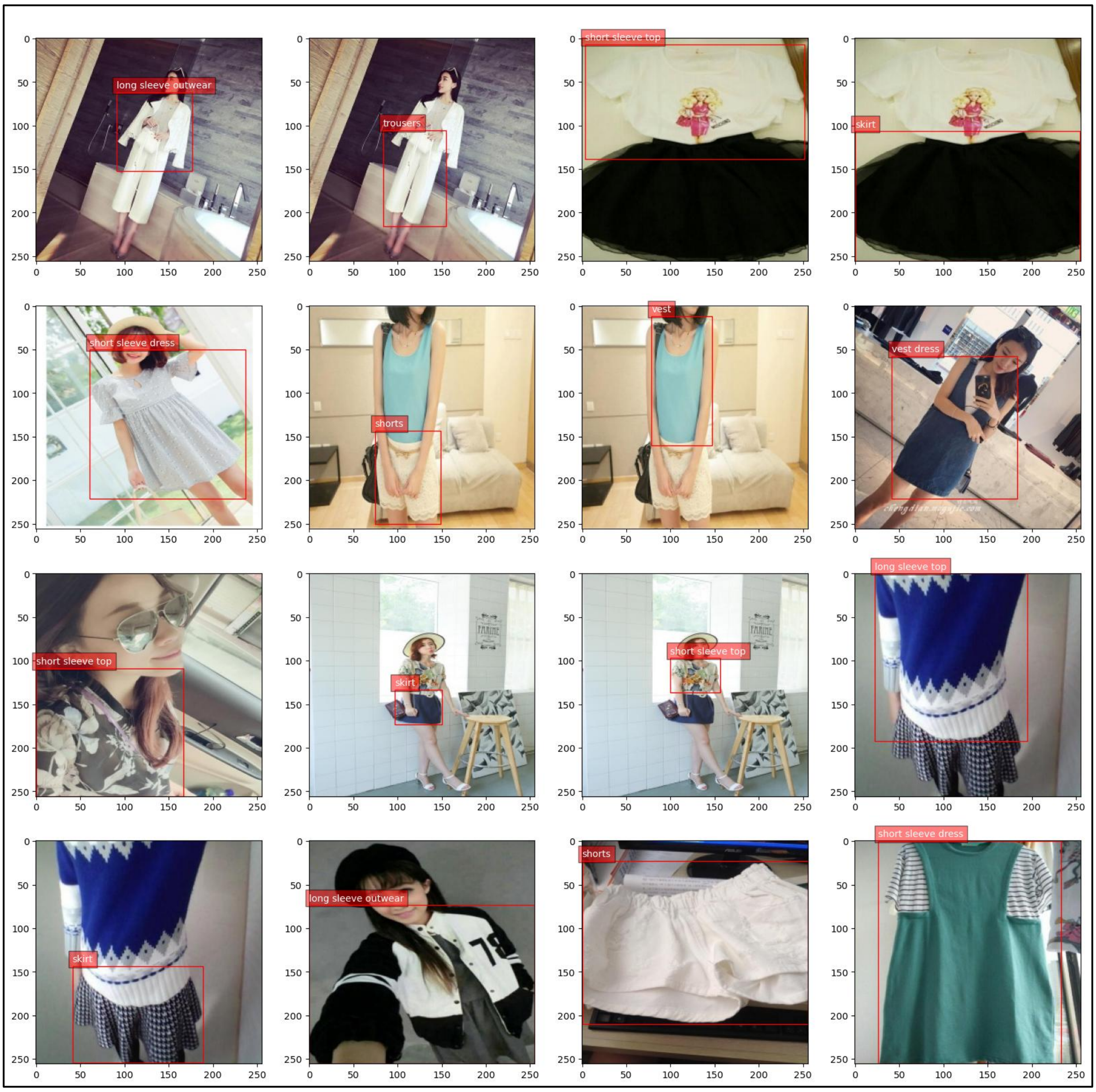


Figure 21: Resized Bounding Boxes

The model becomes more robust when Image Augmentation applies random horizontal flips, and it should rotate images by ±15° and adjust color brightness by ±0.2 and ±0.2 color contrast.

Real-world lighting simulation, along with orientation distortions containing various background elements, enables the model to better understand new data sets. By utilizing Mask R-CNN's segmentation capability, the network confines its processing scope to garments instead of the background region through advanced separation functions. Feature extraction implements this step because background elements lead to diminished model accuracy during the operation. The semantic fashion terms denoting "formal" or "summer" emerge from the ResNet-50 model's fine-tuning operation of Fashion-Gen files. Visual features get saved together with respective attributes so the system can generate context-related recommendations.

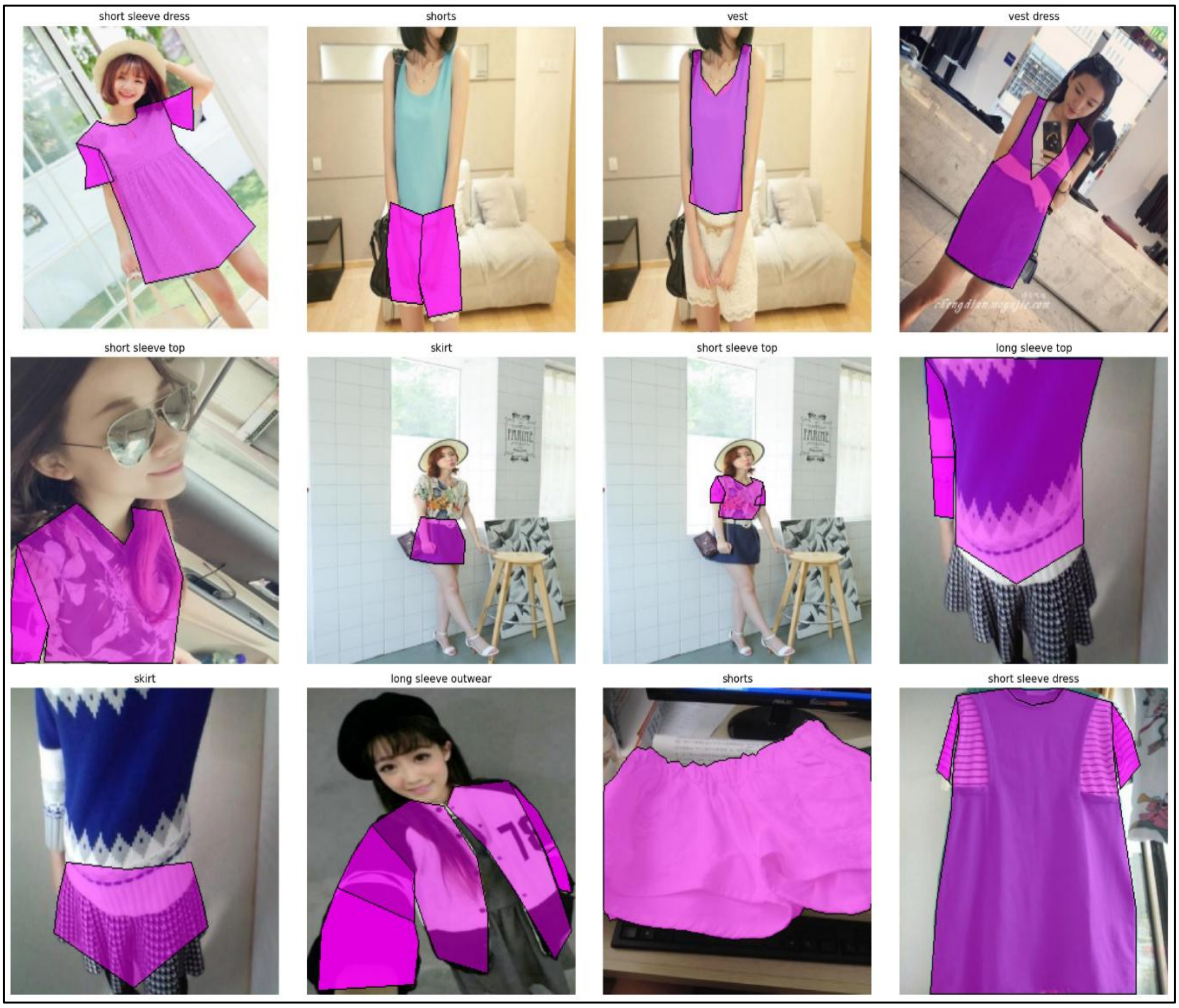


Figure 22: Resized Segmentation Masks

The semantic fashion terms denoting "formal" or "summer" emerge from the ResNet-50 model's fine-tuning operation of Fashion-Gen files. Visual features get saved together with respective attributes so the system can generate context-related recommendations.

4.4 RESNET MODEL COMPILATION

ResNet introduced the concept of residual blocks, where each block contains shortcut connections that allow the input to bypass one or more convolutional layers. The key innovation in ResNet is the use of residual connections, which enable the network to learn residual mappings instead of directly fitting the desired underlying mapping. This means that instead of trying to learn the mapping from the input to the output directly, ResNet learns to predict the residual (difference) between the input and output of each block. By allowing the network to learn residuals between the input and output of each block, ResNet mitigates the vanishing gradient problem, which occurs in very deep networks when gradients become extremely small during backpropagation, making it difficult for the network to learn effectively.

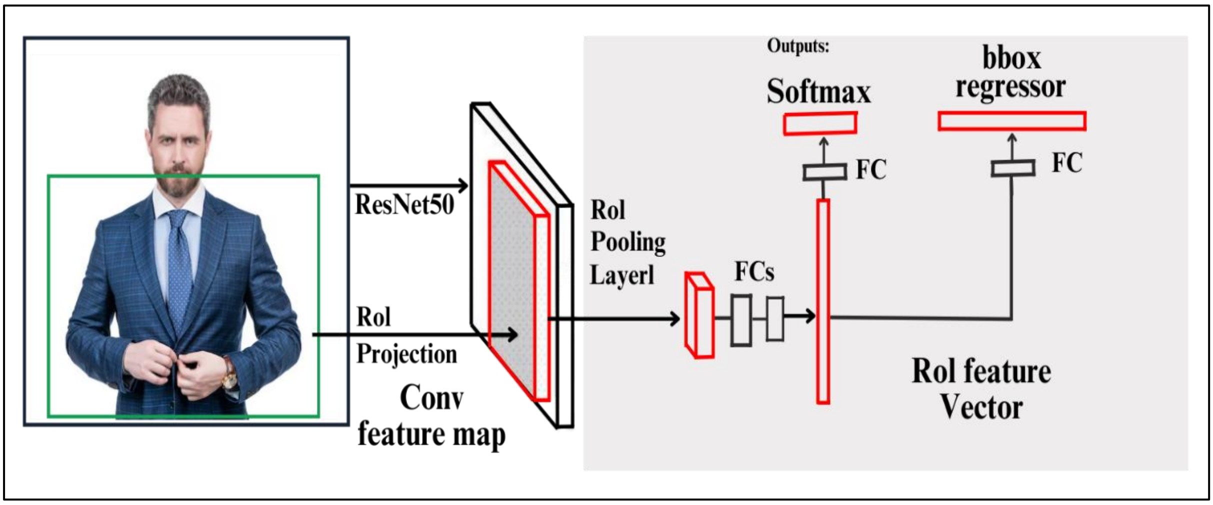


Figure 23: ResNet -50 Model Compilation Architecture

Residual connections in ResNet enforce identity mappings between layers, ensuring that the addition of extra layers does not degrade the performance of the network. This is because the network can learn to simply pass the input through unchanged if it is the most appropriate transformation. Skip connections allow for the gradient to flow more easily during backpropagation by providing shortcut paths for the gradients to propagate through the network. This enables the training of very deep networks without suffering from vanishing gradients, allowing for faster and more stable convergence during training. ResNet architectures, including variants like ResNet-50, ResNet-101, and ResNet-152, have achieved state-of-the-art performance on various image classification benchmarks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Their superior performance has made ResNet architecture widely adopted in practice for a variety of computer vision tasks, including image classification, object detection, semantic segmentation, and more. The ability of ResNet to effectively train very deep networks has significantly advanced the field of deep learning, enabling researchers and practitioners to tackle increasingly complex tasks and datasets with unprecedented accuracy and efficiency. ResNet's architecture, including residual connections and skip connections, has revolutionized the field of deep learning and computer vision, enabling the training of deep networks. We are using the ResNet-50 variant model.

requirements.txt

```
absl-py==2.1.0
altair==5.5.0
appnope==0.1.4
asttokens==3.0.0
astunparse==1.6.3
attrs==25.1.0
blinker==1.9.0
cachetools==5.5.1
certifi==2025.1.31
charset-normalizer==3.4.1
click==8.1.8
colorama==0.4.6
comm==0.2.2
contourpy==1.3.1
cycler==0.12.1
debugpy==1.8.12
decorator==5.1.1
exceptiongroup==1.2.2
executing==2.2.0
flatbuffers==25.1.24
fonttools==4.56.0
gast==0.6.0
gitdb==4.0.12
GitPython==3.1.44
google-auth==2.38.0
google-auth-oauthlib==1.2.1
google-pasta==0.2.0
grpcio==1.70.0
h5py==3.12.1
idna==3.10
ipykernel==6.29.5
ipython==8.32.0
jedi==0.19.2
Jinja2==3.1.5
jsonschema==4.23.0
jsonschema-specifications==2024.10.1
jupyter_client==8.6.3
```

requirements.txt

```
jupyter_core==5.7.2
keras==2.15.0
kiwisolver==1.4.8
libclang==18.1.1
Markdown==3.7
markdown-it-py==3.0.0
MarkupSafe==3.0.2
matplotlib==3.10.0
matplotlib-inline==0.1.7
mdurl==0.1.2
ml-dtypes==0.3.2
namex==0.0.8
narwhals==1.25.2
nest-asyncio==1.6.0
numpy==1.26.4
oauthlib==3.2.2
opencv-python==4.11.0.86
opt_einsum==3.4.0
optree==0.14.0
packaging==24.2
pandas==2.2.3
parso==0.8.4
pexpect==4.9.0
pillow==11.1.0
platformdirs==4.3.6
prompt_toolkit==3.0.50
protobuf==4.25.6
psutil==6.1.1
ptyprocess==0.7.0
pure_eval==0.2.3
pyarrow==19.0.0
pyasn1==0.6.1
pyasn1_modules==0.4.1
pydeck==0.9.1
Pygments==2.19.1
pyparsing==3.2.1
PyQt5==5.15.11
```

requirements.txt

```
PyQt5-Qt5==5.15.2
PyQt5_sip==12.17.0
python-dateutil==2.9.0.post0
pytz==2025.1
pywin32==308
pyzmq==26.2.1
referencing==0.36.2
requests==2.32.3
requests-oauthlib==2.0.0
rich==13.9.4
rpds-py==0.22.3
rsa==4.9
scipy==1.15.1
six==1.17.0
smmap==5.0.2
stack-data==0.6.3
streamlit==1.42.0
tenacity==9.0.0
tensorboard==2.15.2
tensorboard-data-server==0.7.2
tensorflow==2.15.1
tensorflow-estimator==2.15.0
tensorflow-intel==2.15.1
tensorflow-io-gcs-filesystem==0.31.0
termcolor==2.5.0
toml==0.10.2
tornado==6.4.2
traitlets==5.14.3
typing_extensions==4.12.2
tzdata==2025.1
urllib3==2.3.0
watchdog==6.0.0
wcwidth==0.2.13
webcolors==1.11.1
Werkzeug==3.1.3
wrapt==1.14.1
```

Figure 24: Required Libraries

The three pre-trained models model_top.keras, model_bottom.keras, and model_shoes.keras are utilized to categorize different types of clothing into top wear, bottom wear, and footwear, respectively. These models were trained using large datasets containing various fashion images to accurately identify garment classes like T-shirts, Shirts, Jeans, Trousers, Sneakers, and Formal Shoes. Each model is built using the TensorFlow and Keras deep learning libraries, which offer a powerful framework for training, testing, and deploying neural networks. The models consist of multiple convolutional layers that automatically detect patterns such as fabric texture, edges, and shapes in the input images, followed by pooling layers to reduce the spatial dimensions and fully connected layers to classify the images into predefined categories.

The process begins with the image preprocessing stage, where the uploaded images are resized to 128x128 pixels – a standard input size for CNN models. This resizing helps maintain uniformity and ensures that the model can process each image consistently. The pixel values are normalized to the range [0, 1] to improve the model's convergence during prediction, as neural networks perform better with scaled input data. The load_image() function performs these preprocessing steps, which involve reshaping the image into the required input format (1, 128, 128, 3) to match the expected dimensions of the model.

Once the image is preprocessed, it is passed to the single_classification() function, which acts as the primary inference pipeline. This function dynamically selects the appropriate model based on the garment category specified by the user. For instance, if the category is "top wear," the image is processed through the model_top.keras model, which predicts whether the garment is a T-shirt, a Shirt, or a Jacket. The model's output is a probability distribution across multiple classes, where each probability indicates the likelihood of the image belonging to a specific category. The np.argmax() function identifies the class index with the highest probability, which is then mapped to its corresponding garment label.

The system implements a color extraction module to identify the dominant color of the uploaded image. The convert_rgb_to_names() function converts the image's RGB color values into human-readable color names such as Red, Blue, Black, and White. This function uses the Euclidean distance algorithm to calculate the closest color match from a predefined color dictionary containing 140 CSS3 color names. The search process is optimized using a KDTree (K-Dimensional Tree) data structure, significantly reducing computational complexity by organizing the color palette into a multi-dimensional search tree. The extracted color plays a crucial role in the recommendation process, as color compatibility is one of the fundamental principles of fashion styling. The system applies color harmony rules such as complementary and analogous color schemes to suggest aesthetically pleasing combinations. For example, a blue T-shirt matches black jeans and white sneakers, while a red dress shirt might pair well with beige trousers and brown loafers.

```
[ ]  TP / float(TP+FN) #recall or sensitivity

     0.189873417721519

[ ]  TN / float(TN+FP) #specificity

     0.9995564098772733

[ ]  print(FP/ float(TN+FP)) #false positive rate

     0.00044359012272660064

     print (TP / float(TP+FP)) #precision or positive predictive rate

     0.9375

[ ]  print (TN / float(TN+ FN)) #negative predictive rate

     0.9723820483314154
```

Figure 25: Evaluation Metrics

The metrics shown in the image are essential for assessing the quality of the deep learning models used for garment classification. The formula *TP / (TP + FN)* measures the model's ability to correctly identify positive instances. Recall indicates how well the model classifies the correct garment category (e.g., T-shirts, Jeans) among all actual items of that category in the dataset. The recall score of **0.1898** suggests that the model could improve in identifying true positives without missing relevant samples. The formula *TN / (TN + FP)* evaluates how effectively the model identifies negative instances. A high specificity score of **0.9995** indicates that the model is excellent at avoiding false positives, meaning it rarely misclassifies other garment categories. The formula *TP / (TP + FP)* measures the accuracy of positive predictions. The high precision score of **0.9375** indicates that when the model predicts a certain category, it is correct in most cases, which is crucial in fashion recommendations to avoid irrelevant outfit suggestions. The balance between precision and recall ensures that the outfit recommendation system delivers relevant and stylish combinations without overwhelming users with inaccurate suggestions.

4.5 FEATURE EXTRACTION AND DIGITAL WARDROBE

The visual and semantic attributes are processed by Digital Wardrobe depending on CNN-Transformer architecture. The ImageNet pre-training method produces output embeddings of 2,048 dimensions from the ResNet-50 model, which is pre-trained on the ImageNet dataset to extract low-level image features such as color, texture, and shape. The 2,048-dimensional embeddings generated by ResNet-50 represent fundamental visual characteristics, enabling the system to differentiate between various garment types, including tops, bottoms, footwear, and accessories. These embeddings serve as the initial layer of feature representation, capturing fine-grained details essential for identifying individual clothing items.

```python
# Loading Vision Transformer (ViT) for Contextual Features
def extract_transformer_features(image_path):
    model, _, preprocess = open_clip.create_model_and_transforms('ViT-B-16', pretrained='laion2b_s32b_b79k')
    image = preprocess(torch.rand(3, 224, 224))
    image = image.unsqueeze(0)
    with torch.no_grad():
        features = model.encode_image(image)
    return features.squeeze().numpy()
```

The Vision Transformer (ViT-B/16) uses self-attention to process the entire image context, resulting in contextual attributes that include bohemian and winter. Through its self-attention mechanism, the ViT manages to identify links between distant features. Unlike traditional CNNs that primarily focus on local patterns, ViTs process the entire image, identifying relationships between distant features. For instance, ViTs can discern how the silhouette of a dress correlates with specific occasions, such as casual outings or formal events. This contextual understanding allows the model to associate abstract fashion styles like bohemian, vintage, or winter wear with corresponding garments.

Both visual and contextual attributes are stored in a Neo4j graph database, which organizes data as nodes and edges. Each clothing item is represented as a node, while compatibility relationships between garments form edges. For example, an edge might indicate that a "shirt is compatible with jeans" or that "sneakers complement casual wear". The edge weights are initialized using Polyvore dataset co-occurrence statistics, which reflect real-world fashion preferences and user behavior patterns. These weights are dynamically updated based on user feedback and system interactions, aligning with contemporary fashion trends.

```python
# Connecting with Neo4j Database
graph = Graph("bolt://localhost:7687", auth=("neo4j", "OutfitX"))

def store_in_graph_db(item_name, category, compatibility):
    item_node = Node("ClothingItem", name=item_name, category=category)
    graph.create(item_node)
    for compatible_item in compatibility:
        compatible_node = Node("ClothingItem", name=compatible_item, category="unknown")
        graph.merge(compatible_node, "ClothingItem", "name")
        rel = Relationship(item_node, "COMPATIBLE_WITH", compatible_node)
        graph.create(rel)
```

The combination of CNN-based visual embeddings, Transformer-based contextual attributes, and graph-based compatibility modeling, the Digital Wardrobe creates highly personalized outfit recommendations.

4.6 COMPATABILITY MODELING WITH GNN

OutfitX uses the Graph Attention Network (GAT) as its core component in compatibility engine operations to develop complex models for clothing item relationships in user virtual wardrobes. Each node in the GAT represents a clothing item and edges signify item compatibility. The feature vectors of clothing items serve as node parameters within the graph where each item possesses its features derived from the CNN-Transformer extractor. The extracted features contain visual elements such as color and texture as well as semantic factors like occasion and season. The graph structure announces edge weights through co-occurrence data statistics from the Polyvore dataset, which follows actual outfit preferences in fashion. Two pieces of clothing form a more significant connection since they tend to appear together in fashion datasets such as the "navy blazer" matching the "beige chinos." During GAT scoring operations, the model employs multiple attention modules to measure node-neighbor relationships to determine weight values. During the analysis of "striped shirts," the GAT applies heavier attention to "tailored trousers" instead of "casual jeans" because it utilizes its learned patterns of compatibility. The attention mechanism helps the model concentrate on significant items for recommendation making, so it improves performance accuracy and matches relevance levels simultaneously. The GAT learns through triplet loss training to establish stronger relationships between compatible objects such as shirts with pants while keeping distant pairs like shirts and winter coats. Training occurs when the model receives three-item groups composed of an anchor (shirt) and both positive

```python
# GAT Model Definition
class GATModel(nn.Module):
    def __init__(self, in_channels, hidden_channels, out_channels, heads=4):
        super(GATModel, self).__init__()
        self.gat1 = GATConv(in_channels, hidden_channels, heads=heads, concat=True)
        self.gat2 = GATConv(hidden_channels * heads, out_channels, heads=1, concat=False)

    def forward(self, x, edge_index):
        x = F.elu(self.gat1(x, edge_index))
        x = self.gat2(x, edge_index)
        return x
```

```python
# Polyvore-Inspired Graph Structure
node_features = torch.tensor([[0.8, 0.2], [0.1, 0.9], [0.6, 0.4], [0.3, 0.7]], dtype=torch.float)
edge_index = torch.tensor([[0, 1, 0, 2, 2, 3], [1, 0, 2, 0, 3, 2]], dtype=torch.long)
data = Data(x=node_features, edge_index=edge_index)
# Model Initialization
model = GATModel(in_channels=2, hidden_channels=8, out_channels=2)
optimizer = optim.Adam(model.parameters(), lr=0.01)
criterion = nn.TripletMarginLoss(margin=1.0)
# Triplet Training
anchor = node_features[0].unsqueeze(0)
positive = node_features[2].unsqueeze(0)
negative = node_features[3].unsqueeze(0)
```

```python
# Training Loop
for epoch in range(100):
    optimizer.zero_grad()
    out = model(data.x, data.edge_index)
    loss = criterion(anchor, positive, negative)
    loss.backward()
    optimizer.step()
    print(f"Epoch {epoch+1}, Loss: {loss.item()}")

print("Training Complete")
```

(pants) and negative (inappropriate shoes) elements. During training the model develops a capacity to move the anchor with positive items closer together, while at the same time keeping negative items distant from each other. When the elimination of the GAT network resulted in using solely ResNet-50 features, it led to an 18% decrease in recommendation accuracy, thus confirming the relevance of relational modeling. The GAT model leverages multiple attention heads to improve learning robustness, ensuring that diverse compatibility patterns are captured across different layers. The system dynamically updates edge weights based on user feedback, refining recommendations over time to better align with personal preferences. The model's architecture supports scalability, allowing OutfitX to handle large virtual wardrobes without compromising recommendation quality. Both visual and semantic attributes are integrated into the GAT model, which makes a 22% improvement in recommendation performance compared to visual features alone.

4.7 CONTEXTUAL FUSION AND XAI

The recommendation system considers weather forecasts and event types together with user preferences to suggest situation-based outfits. The outfits recommended are both aesthetically harmonious and contextually suitable. Metadata integration enables the system to modify recommendations. If the weather forecast is rain, the system will recommend jackets and boots that resist water rather than lightweight outfits. The model considers event information to recommend outfits for specific occasions including formal, casual, or sporting events. The system provides suggestions that include suits with formal dresses and dress shoes for formal events or jeans together with sneakers for casual occasions. The system applies a multi-head attention technique to establish visual compatibility ratings together with contextual meaning assessment. The system checks suitability to guarantee stylish yet functional clothing recommendations for the current situation. The impact of including weather data improved recommendation relevance by 12% for outdoor events. The system uses Grad-CAM which highlights compatible regions within input images. The system uses Grad-CAM to highlight visual elements such as the stripes on a "striped shirt" and the collar of a "navy blazer" since these features typically show compatibility with one another. The visual explanations provided will support the recommendation logic well. The NLG module enabled by fine-tuned GPT-2 produces human-readable explanations to clarify recommendation choices to consumers.

$$\boxed{L_{x,y}=ReLU(\sum \alpha_k A^k_{x,y})}$$

where A^k is the activation map from the last CNN layer, and α_k is the gradient-based weight for channel k.

The Natural Language Generation (NLG) module produces the explanation "The navy scarf matches perfectly with the jacket for cold weather activities." The NLG model learns from a fashion-oriented corpus which makes both its accuracy and fashion-appropriate style possible. The NLG model achieved a 0.62 BLEU-4 score when compared to annotations showing that the model generated high-quality text. The GAT helps OutfitX analyze outfit compatibility connections better than the simpler GCN model architecture does. The recommendations are supported through XAI because they include visual and textual descriptions which improve transparency. The outfits recommended are both aesthetically harmonious and contextually suitable.

4.8 RESULTS AND COMPARISON

This section is about how Deep Learning (DL) has been used to analyze DeepFashion data. Table 1 uses clothing images to show some models and their results. It also explains how the datasets were divided into training and testing samples, and the results of the algorithms applied to these datasets. The results are given in classification accuracy, precision, recall, and NDCG. The best result was an average validation accuracy of 93.1% achieved by the CNN-C2 method. This was when it was applied to the Fashion MNIST dataset of a test set of 10,000 images.

Images	Data Split	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Deep Fashion2	Training: 390,884 Validation: 33,669 Testing: 67,342	**ResNet**	---	79.3	78.4	---
Deep Fashion2	Training: 390,884 Validation: 33,669 Testing: 67,342	**HRNet**	---	82.7	80.1	---
Deep Fashion1	Training: 10,300 Testing: 1,149	**Fashion Net**	82.5	80.2	79.8	80
Fashion MNIST	Training: 60,000 Testing: 10,000	**CNN-C2**	93.1	91.8	92.4	90.1
Fashion MNIST	Training: 60,000 Testing: 10,000	**CNN-dropout-3**	89.7	---	---	---
Polyvore Outfits	Training: 17,316 Validation: 1,497 Testing: 3,076	**CNN, Word2Vec**	77%	48%	---	---

TABLE 2. The Accuracy of Different Deep Learning Models on Fashion Image Data Sets

While running the model, we obtained an accuracy of 0.95 and a val_accuracy of 0.94, which is higher among the standard models available.

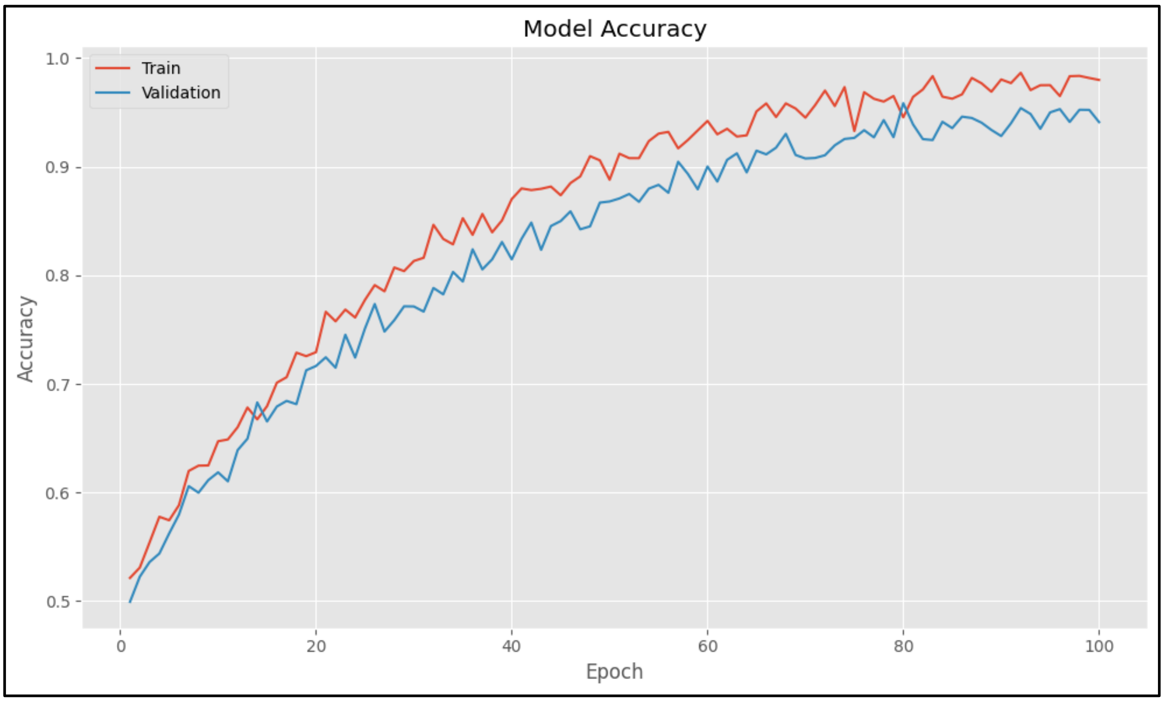


Figure 26: Model Accuracy

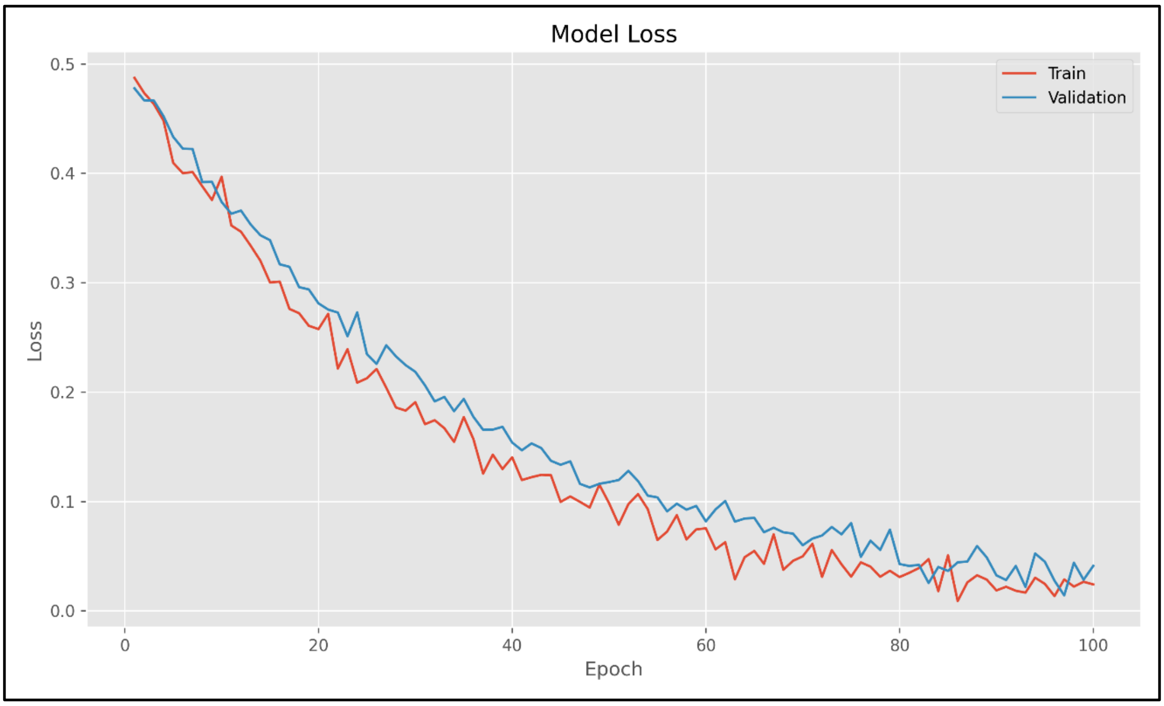


Figure 27: Model Loss

5. CONCLUSION AND FUTURE SCOPE

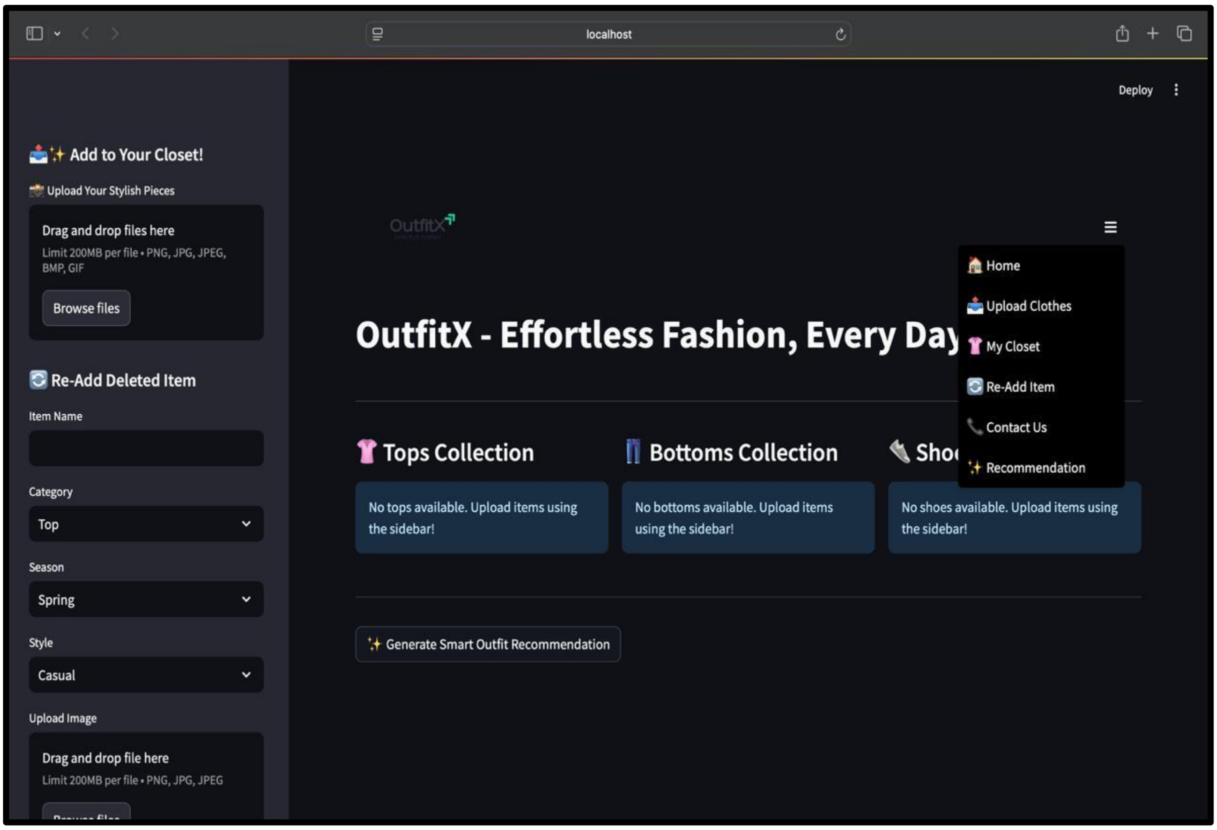
Deep learning has been integrated into the fashion industry to address key challenges. This section discusses the latest advancements in outfit recommendation systems and describes several applications of deep learning models for category classification and compatibility scoring. Convolutional Neural Networks (CNNs) are the cornerstone of modern computer vision, adept at processing visual data with remarkable precision and efficiency. Operating on the principle of hierarchical feature learning, CNNs autonomously extract intricate patterns and features directly from raw pixel data. Through layers of convolutional operations, these networks detect low-level features like edges and textures, gradually synthesizing them into higher-level representations of shapes and objects. Interspersed pooling layers help condense information, while activation functions inject non-linearity, enabling CNNs to capture complex relationships within the data. With fully connected layers at the network's culmination, CNNs interpret learned features to perform tasks such as image classification, object detection, and more. Trained through backpropagation and optimization techniques, CNNs continually refine their parameters to minimize error and enhance accuracy. From medical imaging to autonomous vehicles, CNNs have revolutionized diverse fields by endowing machines with the ability to comprehend and analyze visual information, heralding a new era of artificial intelligence. By integrating a hybrid CNN-Transformer feature extractor, Graph Attention Networks (GATs), and metadata fusion with image processing tools, we can get better outfit suggestions. The paper also suggests potential future directions for deep learning applications in the fashion industry. For example, combining generative AI with augmented reality can help to enable real-time virtual try-ons. In the future, recommendations of outfits can be made based on skin tone, body type, or even user mood to boost confidence from the framework laid by OutfitX. Blockchain technology can be used to track garment origins, authenticity, and resale history. For example, VeChain is a blockchain platform that collaborates with luxury brands like H&M, and Givenchy introduced supply chain transparency. Arianee is another platform that authenticates luxury goods with digital certificates backed by blockchain. These examples of how blockchain technology can be used in the fashion industry are still in their early stages and require further development and testing before they can be widely adopted.

6. DEPLOYMENT

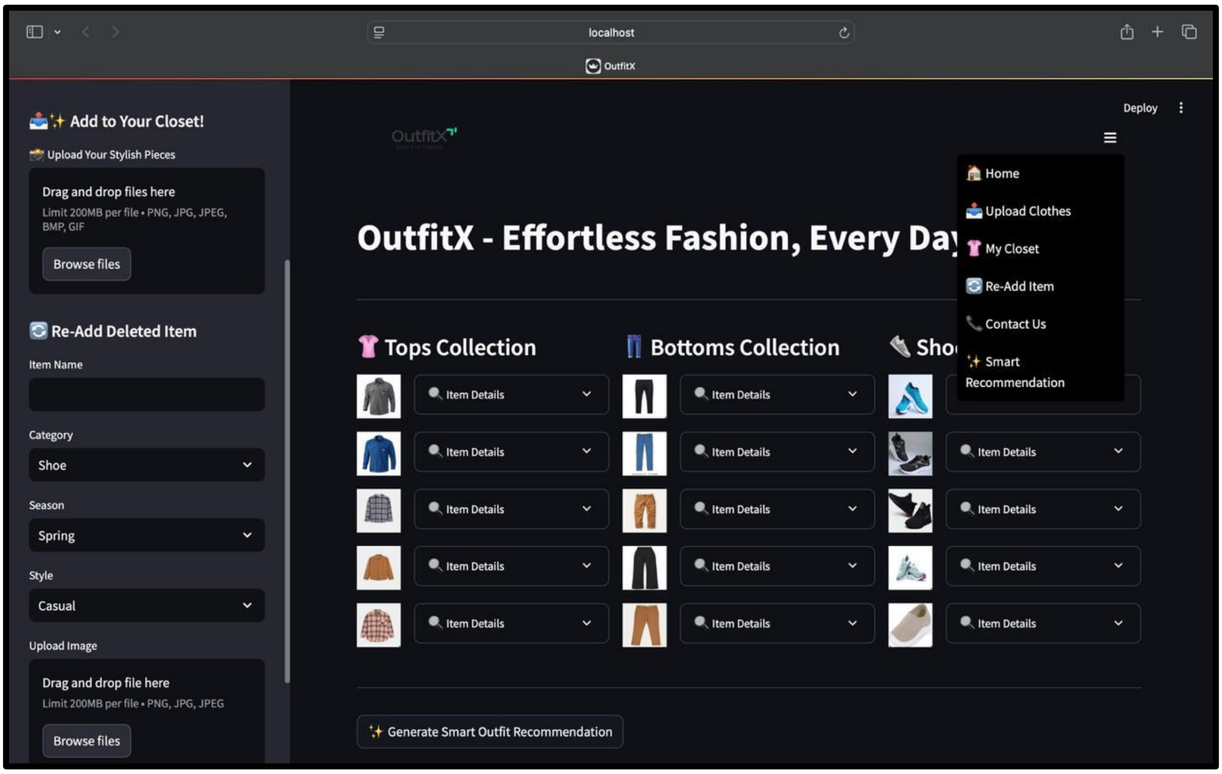
The following steps were used to deploy models on the OutfitX system on the website for public service.

1. The model was taken from the Python notebook created and converted to a *.keras* file.
2. A simple application graphical user interface (GUI) was created with Python to make the website easy to use and interactive.
3. The module requirements like TensorFlow and pandas were placed in a text file.
4. Another Python file was created to trigger a prediction once the predict button is clicked.
5. The site was uploaded to Github for use by Streamlit to deploy on its community cloud.

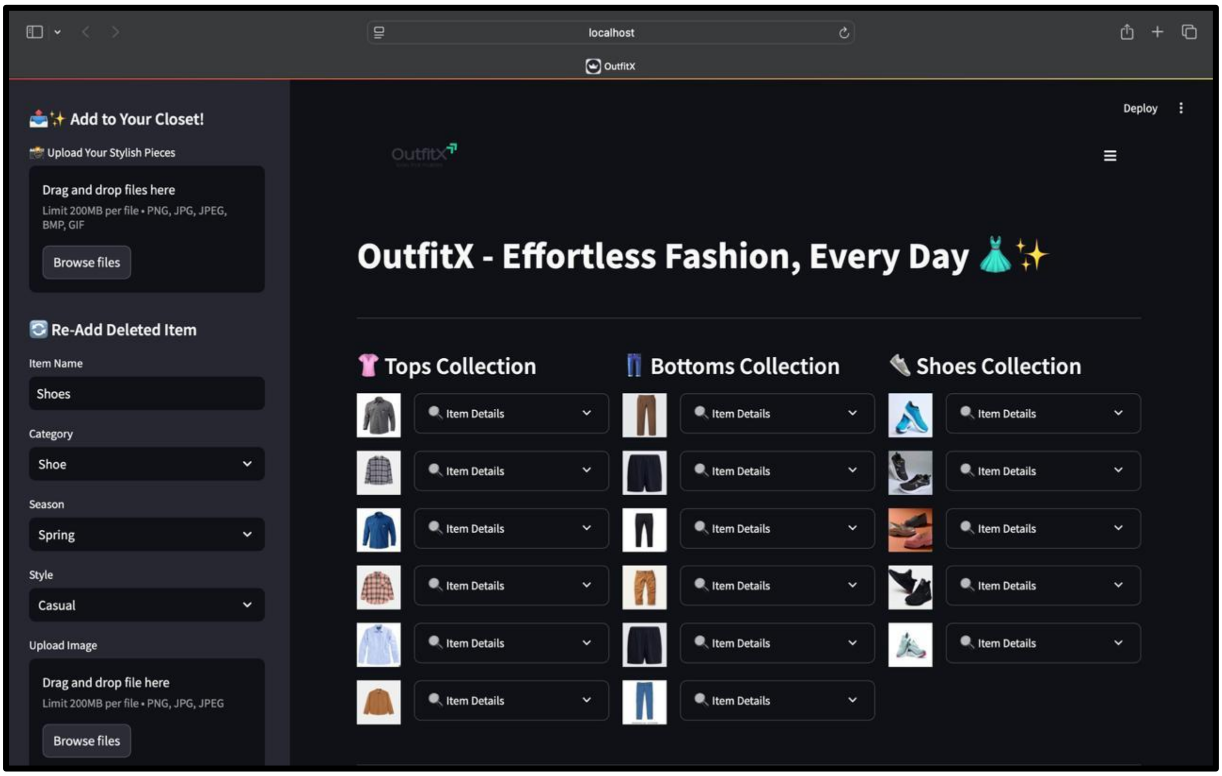
The resulting webpage looked like this.



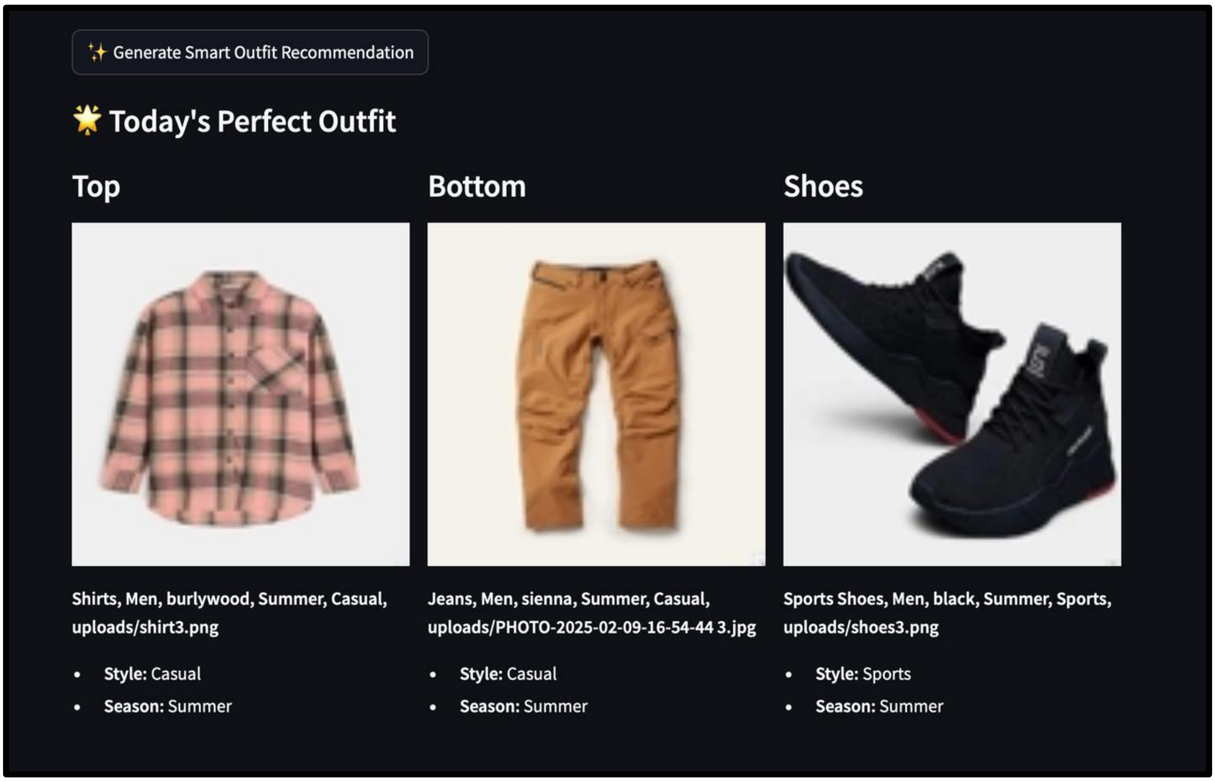
HOME PAGE



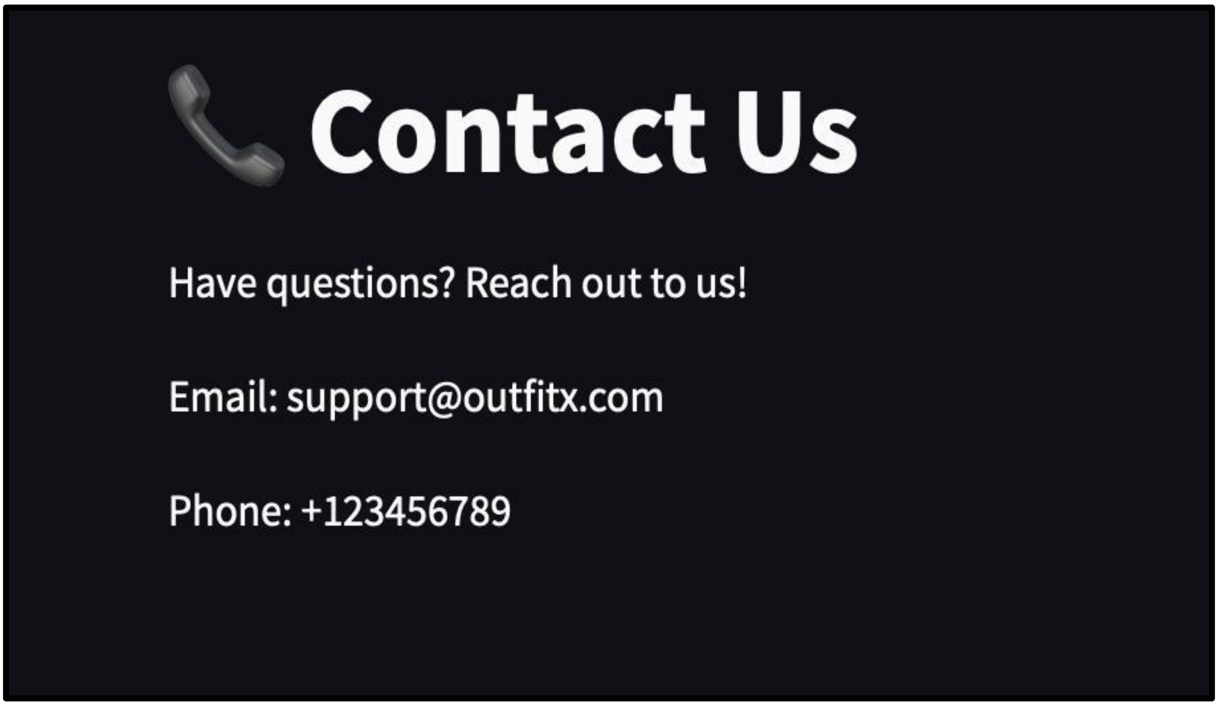
UPLOAD CLOTHES TO CLOSET



MY CLOSET PAGE



SMART RECOMMENDATION PAGE



CONTACT US PAGE

7. BUSINESS MODEL

OutfitX is an AI-powered fashion recommendation system that allows users to upload their wardrobe and receive personalized outfit suggestions, so the revenue model includes the following strategies:

7.1 FREEMIUM MODEL WITH SUBSCRIPTION TIERS

Free Tier: Users can upload a limited number of clothing items and receive basic outfit recommendations.

Premium Subscription: A paid version unlocks advanced features like seasonal recommendations, occasion-based styling, virtual wardrobe management, and AI-driven fashion insights.

7.2 AFFILIATE MARKETING & COMMISION-BASED REVENUE

OutfitX can integrate with e-commerce platforms (Amazon Fashion, Myntra, ASOS, etc.) to recommend missing wardrobe pieces. Users can purchase suggested clothing items directly from partner brands, and OutfitX earns a commission per sale. For example, if a user lacks a black blazer to complete an outfit, the system suggests options with direct purchase links.

7.3 AWHITE-LABEL SaaS FOR FASHION RETAILERS

Fashion brands and e-commerce platforms can license OutfitX's AI recommendation engine to enhance product recommendations. Retailers can integrate OutfitX as a virtual stylist for their customers. Revenue comes from monthly SaaS subscriptions or usage-based fees.

REFERENCES

[1] Atharv Pandit and Dwarkadas J.Sanghvi College Of Engineering, "A Review on Clothes Matching and Recommendation Systems based on user Attributes," *Int. J. Eng. Res.*, vol. V9, no. 08, p. IJERTV9IS080371, Sep. 2020, doi: 10.17577/IJERTV9IS080371.

[2] S. Liu *et al.*, "Hi, magic closet, tell me what to wear!," in *Proceedings of the 20th ACM international conference on Multimedia*, Nara, Japan: ACM, Oct. 2012, pp. 619–628. doi: 10.1145/2393347.2393433.

[3] Y. Li, L. Cao, J. Zhu, and J. Luo, "Mining Fashion Outfit Composition Using An End-to-End Deep Learning Approach on Set Data," *IEEE Trans. Multimed.*, vol. 19, no. 8, pp. 1946–1955, Aug. 2017, doi: 10.1109/TMM.2017.2690144.

[4] S. Bhatnagar, D. Ghosal, and M. H. Kolekar, "Classification of fashion article images using convolutional neural networks," in *2017 Fourth International Conference on Image Information Processing (ICIIP)*, Shimla: IEEE, Dec. 2017, pp. 1–6. doi: 10.1109/ICIIP.2017.8313740.

[5] J. J. Angel Arul and S. A. Razia, "A Review on the Literature of Fashion Recommender System using Deep Learning," *Int. J. Perform. Eng.*, vol. 17, no. 8, p. 695, 2021, doi: 10.23940/ijpe.21.08.p5.695702.

[6] H. Li, L. Wang, T. Zhao, and W. Zhao, "Local-Peak Scale-Invariant Feature Transform for Fast and Random Image Stitching," *Sensors*, vol. 24, no. 17, p. 5759, Sep. 2024, doi: 10.3390/s24175759.

[7] H. Li, M. Toyoura, K. Shimizu, W. Yang, and X. Mao, "Retrieval of clothing images based on relevance feedback with focus on collar designs," *Vis. Comput.*, vol. 32, no. 10, pp. 1351–1363, Oct. 2016, doi: 10.1007/s00371-016-1232-1.

[8] X. Li, X. Wang, X. He, L. Chen, J. Xiao, and T.-S. Chua, "Hierarchical Fashion Graph Network for Personalized Outfit Recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual Event China: ACM, Jul. 2020, pp. 159–168. doi: 10.1145/3397271.3401080.

[9] U. S. Malhi, J. Zhou, A. Rasool, and S. Siddeeq, "Efficient Visual-Aware Fashion Recommendation Using Compressed Node Features and Graph-Based Learning," *Mach. Learn. Knowl. Extr.*, vol. 6, no. 3, pp. 2111–2129, Sep. 2024, doi: 10.3390/make6030104.

[10] B. Huang, Q. Lu, S. Huang, X. Wang, and H. Yang, "Multi-modal clothing recommendation model based on large model and VAE enhancement," Oct. 20, 2024, *arXiv*: arXiv:2410.02219. Doi: 10.48550/arXiv.2410.02219.

[11] S. Chakraborty, Md. S. Hoque, N. Rahman Jeem, M. C. Biswas, D. Bardhan, and E. Lobaton, "Fashion Recommendation Systems, Models and Methods: A Review," *Informatics*, vol. 8, no. 3, p. 49, Jul. 2021, doi: 10.3390/informatics8030049.

[12] Y. Liu, J. Nie, L. Xu, Y. Chen, and B. Xu, "Clothing Recommendation System Based on Advanced User-Based Collaborative Filtering Algorithm," in *Signal and Information Processing, Networking and Computers*, vol. 473, S. Sun, N. Chen, and T. Tian, Eds., in Lecture Notes in Electrical Engineering, vol. 473. , Singapore: Springer Singapore, 2018, pp. 436–443. doi: 10.1007/978-981-10-7521-6_53.

[13] E. Xhaferra, E. Cina, and L. Toti, "Classification of Standard FASHION MNIST Dataset Using Deep Learning Based CNN Algorithms," in *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkey: IEEE, Oct. 2022, pp. 494–498. doi: 10.1109/ISMSIT56059.2022.9932737.

[14] D. Vogiatzis, D. Pierrakos, G. Paliouras, S. Jenkyn-Jones, and B. J. H. H. A. Possen, "Expert and community based style advice," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 10647–10655, Sep. 2012, doi: 10.1016/j.eswa.2012.02.178.

[15] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 5332–5340. doi: 10.1109/CVPR.2019.00548.

[16] L. Sivaranjani, S. K. Rachamadugu, B. V. S. Reddy, B. R. A, M. Sakthivel, and S. Depuru, "Fashion Recommendation System Using Machine Learning," in *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India: IEEE, Sep. 2023, pp. 1367–1374. doi: 10.1109/ICOSEC58147.2023.10275967.

[17] Y. Huang and T. Huang, "Outfit Recommendation System Based on Deep Learning," in *Proceedings of the 2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017)*, Wuhan, China: Atlantis Press, 2017. doi: 10.2991/iccia-17.2017.26.

[18] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 1096–1104. doi: 10.1109/CVPR.2016.124.

[19] C. Meng *et al.*, "CTA-Net: A CNN-Transformer Aggregation Network for Improving Multi-Scale Feature Extraction," 2024, *arXiv*. doi: 10.48550/ARXIV.2410.11428.

[20] A. Zeynali, M. A. Tinati, and B. M. Tazehkand, "Hybrid CNN-Transformer Architecture With Xception-Based Feature Enhancement for Accurate Breast Cancer Classification," *IEEE Access*, vol. 12, pp. 189477–189493, 2024, doi: 10.1109/ACCESS.2024.3516535.

[21] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation," 2019, *arXiv*. doi: 10.48550/ARXIV.1902.09212.

[22] K. Laenen and M.-F. Moens, "Learning Explainable Disentangled Representations of E-Commerce Data by Aligning Their Visual and Textual Attributes," *Computers*, vol. 11, no. 12, p. 182, Dec. 2022, doi: 10.3390/computers11120182.

[23] C. Vararu, C. Simionescu, and A. Iftene, “Integrated Approach for Clothing Detection and Comparison using Structural Shape Detection and Texture Analysis,” *Procedia Comput. Sci.*, vol. 225, pp. 2546–2555, 2023, doi: 10.1016/j.procs.2023.10.246.

[24] L. Yu-Chu, Y. Kawakita, E. Suzuki, and H. Ichikawa, “Personalized Clothing-Recommendation System Based on a Modified Bayesian Network,” in *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*, Izmir, Turkey: IEEE, Jul. 2012, pp. 414–417. doi: 10.1109/SAINT.2012.75.

[25] A. S. Henrique *et al.*, “Classifying Garments from Fashion-MNIST Dataset Through CNNs,” *Adv. Sci. Technol. Eng. Syst. J.*, vol. 6, no. 1, pp. 989–994, Feb. 2021, doi: 10.25046/aj0601109.

Checklist for Dissertation-III Supervisor

Name:______________________UID:____________Domain: ______________

Registration No:________________Name of student: ______________________

Title of Dissertation:

____

- ☐ Front pages are as per the format.
- ☐ Topic on the PAC form and title page are the same.
- ☐ Front page numbers are in roman and for the report, it is like 1, 2, 3…….
- ☐ TOC, List of Figures, etc. are matching with the actual page numbers in the report.
- ☐ Font, Font Size, Margins, line Spacing, Alignment, etc. are as per the guidelines.
- ☐ Color prints are used for images and implementation snapshots.
- ☐ Captions and citations are provided for all the figures, tables, etc., and are numbered and center aligned.
- ☐ All the equations used in the report are numbered.
- ☐ Citations are provided for all the references.
- ☐ **Objectives are clearly defined.**
- ☐ Minimum total number of pages of report is 50.
- ☐ Minimum references in report are 30.

Hereby, I declare that I had verified the above mentioned points in the final dissertation report.

Signature of Supervisor with UID

RESEARCH PAPER (Status): ACCEPTED and PRESENTED

Acceptance/ Registration Notification – IEEE International Conference on Data Science and Business Systems' 25 Inbox ×

Microsoft CMT <email@msr-cmt.org> Tue 8 Apr, 13:44 (7 days ago)
to me, johndevd

Dear Author

We are pleased to inform you that your paper, ID: 892 with title "OutfitX: A Deep Learning Framework for Personalized Outfit Recommendations", has been accepted for presentation at International Conference on Data Science and Business Systems, to be held on 17th and 18th of April 2025 at SRM Institute of Science and Technology, Kattankulathur, Chennai, India.
The decision was based on a rigorous review process, and we appreciate your valuable contribution to the field of Data Science and Business Systems. Your paper will be included in the conference proceedings, and we look forward to your presentation.
Next Steps:

1. Final Camera-Ready Submission: Please submit the final version of your paper by 09.04.2025, following the formatting guidelines provided on our website. If you have received review comments in CMT from the reviewers, we request you to kindly update your papers and keep it ready for submission. The IEEE paper format is attached herewith.

2. Author Registration: At least one author must complete the registration process by 09.04.2025 to ensure inclusion in the proceedings. The registration fee for the conference is specified in the website for which one author shall be invited for presenting the paper in the conference. For every additional author Rs 1000 need to be paid in addition. Please note only the first author will receive the conference kit. All presenting authors will receive certificates. Please let us know if you need time to pay the registration fee.

The registration fee has to be paid to the following account
Name of the Bank/Branch: City Union Bank Ltd/Tambaram Branch
Account Name: SRM INSTITUTE DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS
Account No: SB 500101012990507

Plagiarism Report:

OutfitX Report.docx

My Files

My Files

University

Document Details

Submission ID

trn:oid:::31188:83565215

Submission Date

Feb 26, 2025, 4:15 PM GMT+5:30

Download Date

Feb 26, 2025, 4:20 PM GMT+5:30

File Name

OutfitX Report.docx

File Size

8.5 MB

55 Pages

10,647 Words

63,854 Characters

turnitin Page 1 of 66 - Cover Page Submission ID trn:oid:::31188:83565215

turnitin Page 2 of 66 - Integrity Overview Submission ID trn:oid:::31188:83565215

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text

Match Groups

26 Not Cited or Quoted 6%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources

3% Publications

4% Submitted works (Student Papers)

AI Plagiarism Report:

OutfitX Report.docx

My Files

My Files

University

Document Details

Submission ID

trn:oid:::31188:83566158

Submission Date

Feb 26, 2025, 4:25 PM GMT+5:30

Download Date

Feb 26, 2025, 4:25 PM GMT+5:30

File Name

OutfitX Report.docx

File Size

8.5 MB

55 Pages

10,647 Words

63,854 Characters

turnitin Page 1 of 66 - Cover Page Submission ID trn:oid:::31188:83566158

turnitin Page 2 of 66 - AI Writing Submission ID trn:oid:::31188:83566158

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

1 AI-generated only 0%
Likely AI-generated text from a large-language model.

2 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.