Smart Wardrobe: A Comprehensive Approach to Personalized Clothing Recommendation with the use of Nearest Neighbor Model

Vidit Parekh
Artificial Intellegence & Machine Learning
Shri Dwarkadas J. Sanghvi College of Engineering

Yash Shah
Artificial Intellegence & Machine Learning
Shri Dwarkadas J. Sanghvi College of Engineering

Dr. Aruna Gawade
Artificial Intellegence & Machine Learning
Shri Dwarkadas J. Sanghvi College of Engineering

Dr. Nilesh Rathod
Artificial Intellegence & Machine Learning
Shri Dwarkadas J. Sanghvi College of Engineering

Prof. Komal Patil
Artificial Intellegence & Machine Learning
Shri Dwarkadas J. Sanghvi College of Engineering

Abstract

This research empowers users to manage their wardrobe and create stylish outfits using image recognition and machine learning and it focuses on outfit creation. Users upload photos of their existing garments, and the application utilizes K-Nearest Neighbors (KNN) to analyze the visual features of each item (color, style, pattern). When a user selects an occasion (work, party, etc.), Smart Wardrobe leverages KNN to recommend outfit combinations by finding similar or complementary items within their wardrobe. For occasions where the user lacks suitable clothing, Smart Wardrobe seamlessly integrates e-commerce functionality. By analyzing the chosen occasion and the user's wardrobe composition, the application recommends outfit components the user might be missing. These recommendations include links to relevant online stores, allowing users to effortlessly complete their desired look. This innovative approach transcends simple outfit suggestions, transforming Smart Wardrobe into a personalized fashion assistant that optimizes wardrobe utilization and streamlines the clothing selection process.

Introduction

The realm of fashion presents both a canvas for self-expression and a potential source of daily frustration. Choosing the perfect outfit requires navigating a sea of garments, considering factors like occasion, personal style, weather, and current trends. This decision-making process can be particularly time-consuming and overwhelming for individuals with busy schedules or limited fashion knowledge. This research proposes "Smart Wardrobe," a groundbreaking mobile application designed to revolutionize the way users approach clothing selection. Leveraging image recognition and machine learning, Smart Wardrobe empowers users to manage their existing wardrobes and curate stylish outfits tailored to specific occasions. Moving beyond traditional size prediction applications, Smart Wardrobe focuses on maximizing the potential of a user's existing clothing collection. The process of outfit creation involves several challenges. Firstly, managing a personal wardrobe can be cumbersome. Individuals often struggle to keep track of all their clothing items, leading to forgotten garments and underutilized pieces. Secondly, the sheer volume of clothing choices can be overwhelming, leading to decision fatigue and frustration. Finally, aligning personal style with appropriate attire for specific occasions necessitates knowledge of current trends, dress codes, and weather conditions. These factors combine to create a complex decision-making process for many individuals. Smart Wardrobe addresses these challenges by offering a comprehensive wardrobe management and outfit recommendation system. The core functionalities of the application can be summarized as follows:

a. **Wardrobe Management:** Utilizing image recognition technology, Smart Wardrobe allows users to upload photos of their clothing items. The application then meticulously analyzes these images, categorizing each garment (shirts, pants, dresses, etc.) and extracting key details like color, style, pattern, and brand (if applicable). This process creates a digital inventory of a user's wardrobe, facilitating better organization and eliminating the need to physically sort through garments.

b. **K-Nearest Neighbors (KNN) for Outfit Recommendations:** When a user selects an occasion (work, date night, etc.), Smart Wardrobe employs the K-Nearest Neighbors (KNN) algorithm to recommend outfit combinations. KNN analyzes the visual features (color, pattern, style) of a user-selected garment (base image) and identifies the "K" nearest neighbors (similar items) within their wardrobe. By recommending these visually complementary pieces, Smart Wardrobe facilitates the creation of cohesive and stylish outfits.

c. **E-commerce Integration for Outfit Completion:** While Smart Wardrobe prioritizes maximizing wardrobe utilization, it acknowledges that some occasions may require specific clothing items that a user may not currently own. To address this need, Smart Wardrobe seamlessly integrates e-commerce functionality. The application analyzes the chosen occasion and the user's wardrobe composition. If a crucial piece is missing, Smart Wardrobe recommends the missing outfit components along with links to relevant online stores. This allows users to effortlessly complete their desired look without leaving the app.

The Significance of Smart Wardrobe:

Smart Wardrobe transcends the realm of a typical fashion app by offering a multitude of benefits to users. It fosters improved wardrobe organization, simplifies outfit creation, and streamlines the clothing selection process. Furthermore, the application empowers users to develop a deeper understanding of their personal style by encouraging them to explore different outfit combinations within their existing wardrobe. This fosters a sense of self-confidence and reduces reliance on impulse purchases. Additionally, Smart Wardrobe's e-commerce integration offers a convenient solution for acquiring missing pieces and completing an outfit. This research delves into the technical details of Smart Wardrobe, exploring the image recognition algorithms employed for wardrobe organization, the KNN approach for outfit recommendation, and the integration of e-commerce functionalities. Through a thorough evaluation process, the paper will assess the effectiveness of Smart Wardrobe's core functionalities and its potential impact on user experience and fashion choices. The potential societal impact of Smart Wardrobe is significant. By promoting efficient wardrobe management and reducing reliance on impulsive clothing purchases, the application can contribute to a more sustainable fashion ecosystem. Furthermore, Smart Wardrobe empowers individuals with limited fashion knowledge or busy schedules to navigate the world of fashion with greater confidence and ease. Ultimately, Smart Wardrobe paves the way for a future where technology seamlessly integrates with daily life, simplifying daily routines and transforming the way users interact with their wardrobes and the broader fashion landscape.

II. LITERATURE SURVEY

The research paper titled "Smart clothing recommendation system with deep learning" by Aşıroğlu et al. (2019) presents a novel approach to clothing recommendation leveraging deep learning techniques. While the paper primarily focuses on the development and implementation of the recommendation system, it builds upon existing literature in several key areas. Firstly, it draws upon the extensive research in the field of recommendation systems, particularly those utilizing machine learning and deep learning algorithms for personalized recommendations. Secondly, it incorporates studies related to smart textiles and wearable technology, highlighting the growing interest in integrating technology into clothing for various applications. Additionally, the paper likely references literature exploring the intersection of fashion and technology, as the recommendation system's goal is to provide personalized clothing suggestions. By synthesizing insights from these domains, the research contributes to the advancement of intelligent systems for enhancing user experiences in the fashion industry.C. Published in 2022, the research paper "Human Face Shape Classification with Machine Learning" by Ashwinee Mehta and Taha Mahmoud explores the utilisation of machine learning techniques for the precise categorization of human face shapes. The primary objective is to create an automated system that effectively classifies individuals' face shapes, contributing to advancements in facial recognition and analysis. Employing machine learning algorithms, the authors construct a dataset comprising annotated facial images for training and evaluating the model. They experiment with diverse feature extraction techniques and algorithms to enhance the model's capacity to discern distinctive characteristics associated with various face shapes. Classification accuracy serves as a key metric for assessing the proposed approach, with the authors addressing challenges like lighting variations and facial expressions. Mehta and Mahmoud's work underscores the significance of advanced computational methods in comprehending and applying facial features across diverse practical domains.

The paper by Liang (2020) titled "Image classification based on RESNET" likely draws from a comprehensive literature survey to establish the foundation for its work. It would likely encompass various aspects of image classification and deep learning techniques. Firstly, it might reference seminal works in the field of image classification, such as AlexNet, VGG, and GoogLeNet, to provide context and comparison for the ResNet architecture. Additionally, the literature survey may explore the development and advancements in convolutional neural networks (CNNs), particularly focusing on the evolution of deeper architectures like ResNet. Furthermore, it might delve into the applications of ResNet in various domains beyond image classification, such as object detection and semantic segmentation, to showcase its versatility and effectiveness. By synthesizing insights from these sources, the paper sets the stage for its contribution, which likely involves proposing enhancements or optimizations to the ResNet architecture for improved image classification performance.

The paper by Guan et al. (2016) titled "Apparel recommendation system evolution: an empirical review" provides a comprehensive literature survey on the evolution of apparel recommendation systems. It likely encompasses various facets of research in this domain. Firstly, it would explore early approaches to apparel recommendation, potentially dating back to rule-based systems or collaborative filtering techniques. Secondly, the survey would likely cover advancements in recommendation algorithms, such as content-based filtering, collaborative filtering, and hybrid methods, discussing their strengths and limitations in the context of apparel recommendation. Additionally, the paper might delve into the integration of emerging technologies like machine learning, deep learning, and natural language processing into apparel recommendation systems to improve accuracy and personalization. Furthermore, it could explore the impact of factors such as user preferences, context-aware recommendations, and social influence on the effectiveness of apparel recommendation systems. By synthesizing insights from these sources, the paper aims to provide a comprehensive understanding of the evolution and current state of apparel recommendation systems.

The paper by Shin et al. (2023) titled "A novel method for fashion clothing image classification based on deep learning" likely incorporates a literature survey that provides a comprehensive overview of relevant research in the field of fashion image classification and deep learning. Firstly, it would likely discuss seminal works in image classification, particularly those focused on fashion datasets, such as Fashion-MNIST or DeepFashion, to establish a foundational understanding of the field. Secondly, the survey may explore various deep learning architectures commonly used for image classification tasks, including CNNs like VGG, ResNet, and Inception, as well as more recent advancements in architecture design. Additionally, it might discuss existing methodologies and techniques for feature extraction, data augmentation, and model optimization specific to fashion image classification. Furthermore, the survey could touch upon applications of deep learning in fashion beyond image classification, such as style recommendation, trend analysis, and virtual try-on systems. By synthesizing insights from these sources, the paper aims to present a novel approach that contributes to the advancement of fashion image classification techniques.

The paper by Liang (2020) titled "Image classification based on RESNET" likely incorporates a literature survey that provides an overview of key research in the fields of image classification and deep learning architectures. Firstly, it may explore seminal works in image classification, such as AlexNet, VGG, and GoogLeNet, to establish the evolution of deep learning models for this task. Secondly, the survey may delve into the development of ResNet and its variants, discussing their architectural innovations and performance improvements over earlier models. Additionally, it might discuss methodologies for training deep neural networks effectively, including techniques like transfer learning and data augmentation. Furthermore, the survey could touch upon applications of ResNet beyond image classification, such as object detection and semantic segmentation. By synthesizing insights from these sources, the paper aims to contribute to the understanding of ResNet's effectiveness and potential applications in image classification tasks.

III. METHODOLOGY

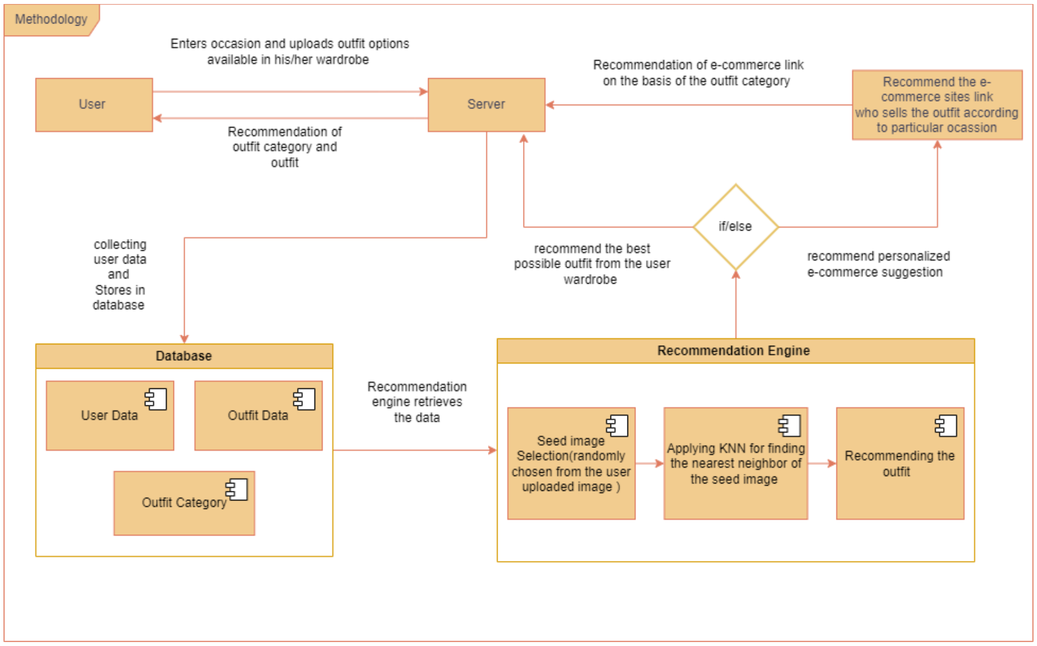

Fig 3.1 Architecture diagram of the Smart Wardrobe

In the above **Fig 3.1** it depicts an architecture diagram outlining the recommendation engine architecture leverages a user's existing wardrobe to generate outfit recommendations. It employs machine learning techniques to identify similar clothing items and builds outfits based on those similarities. The system also appears to have the capability to suggest items from external retailers if suitable options are not found within a user's wardrobe.

User: This is the starting point of the recommendation process. The user interacts with the system by entering details regarding an occasion (such as "work", "casual", or "formal") and uploading photos of various outfit options available in their wardrobe. **Database:** This is a repository that stores information relevant to the recommendation process. It can be broken down into two parts:

a. **User Data:** This likely includes a unique user ID, any preferences the user has specified (such as favourite colours or styles), and possibly a history of past interactions with the system, which could include past occasions entered and past outfit selections.
b. **Outfit Data:** This likely includes details on items in the user's wardrobe, including descriptions, categories (such as tops, bottoms, dresses, shoes, or accessories), colours, brands, and possibly images of the clothing items.

Server: This component acts as an intermediary between the user and the recommendation engine. It receives user requests, including occasion details and outfit image uploads. It stores this information in the database's user data and outfit data sections. The server then retrieves data from the database to facilitate the recommendation process by providing the recommendation engine with the necessary information. **Recommendation Engine:** This is the core functionality of the system and is responsible for generating outfit recommendations tailored to the user's input and wardrobe. It retrieves data from the database, including details about the occasion provided by the user and information on the various clothing items in the user's wardrobe.

a. **Seed Image Selection:** The engine utilizes a process to choose an image from the user's uploaded options as a starting point, possibly at random. This seed image acts as a reference point from which to generate outfit recommendations.
b. **KNN (K-Nearest Neighbors) Algorithm:** The engine employs this machine learning technique to find similar clothing items to the seed image within the user's wardrobe. KNN is an instance-based learning algorithm that classifies data points based on their proximity to labeled data points. In the case of this

recommendation engine, it seems to be used to find outfit items within a user's wardrobe that are similar to the chosen seed image.

c. **Recommendation of Outfit Category and Outfit:** Based on the results from the KNN algorithm, the engine generates recommendations that include both outfit categories (e.g., tops, bottoms, dresses) and specific outfit combinations. For instance, if the seed image is a pair of jeans, the recommendation engine might recommend pairing the jeans with a t-shirt and sneakers for a casual occasion.

There is also an **"if/else"** decision point within the recommendation engine. This suggests there may be logic built into the system to handle situations where the KNN algorithm may not generate suitable outfit recommendations from the user's wardrobe. In such cases, the engine might employ a fallback recommendation strategy, such as suggesting similar items from a third-party e-commerce website.

Recommendation of E-commerce Link: This section suggests that the recommendation engine might also provide links to external e-commerce websites. This could be the case if the user's wardrobe lacks certain items to complete a recommended outfit, or if the engine identifies opportunities to suggest complementary items that enhance the recommended outfit. For example, if the recommendation engine suggests a dress but the user doesn't have shoes that would match well, the engine might also recommend shoes from an e-commerce website. The logic behind e-commerce recommendations might consider the occasion, the recommended outfit category (e.g., dress), and potentially user data or past purchase history.

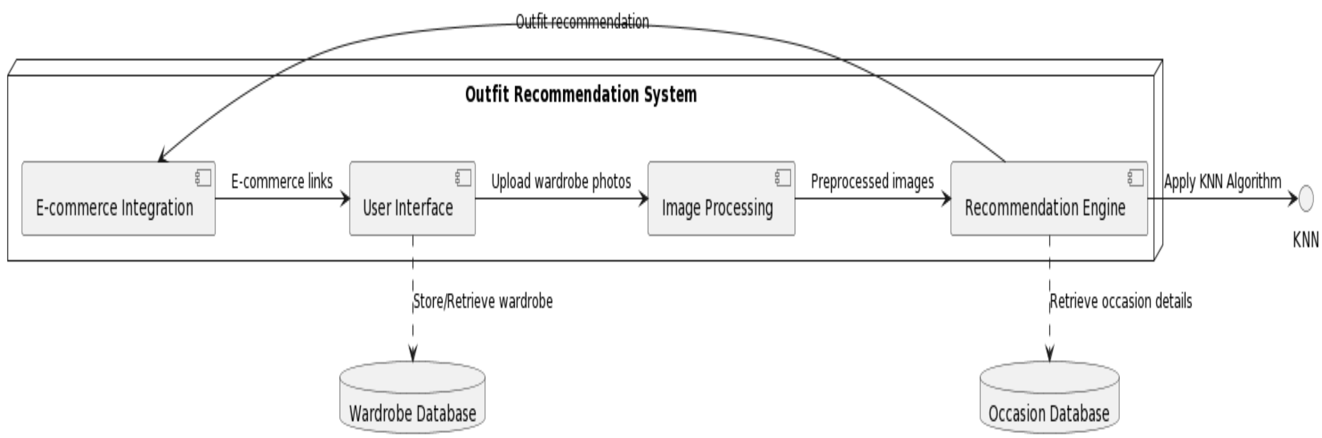

Fig 3.2 Component Diagram for Outfit Recommendation System

The **Fig 3.2** above depicts a block diagram of a cloud recommendation system designed for recommending outfits. Overall, the cloud recommendation system depicted in the block diagram facilitates a personalized outfit recommendation experience by leveraging user wardrobe data, occasion details, and a KNN algorithm to generate outfit suggestions. The system leverages a KNN (K-Nearest Neighbors) algorithm to suggest apparel based on a user's wardrobe and occasion details. Here's a detailed explanation of the cloud recommendation system block diagram:

Upload wardrobe photos: Users can upload photos of their clothing articles into the system. **Preprocessed images:** The system preprocesses the uploaded wardrobe images. Pre-processing likely involves resizing the images, converting them to a standard format, and potentially extracting features like color and patterns from the images. **Store/Retrieve wardrobe:** The preprocessed wardrobe images are stored in a wardrobe database, which can be retrieved when needed. **E-commerce Integration:** The system integrates with e-commerce platforms, which provide information about available outfits. **Occasion Database:** The system stores details about various occasions in an occasion database. **Retrieve occasion details:** Users can provide details about a specific occasion for which they require an outfit recommendation. **Apply KNN Algorithm:** The system utilizes a KNN algorithm to recommend outfits. KNN is a machine learning algorithm that classifies data points by identifying the nearest neighbors (similar data points) according to a specified distance metric. In this context, the KNN algorithm would likely identify similar clothing items in the e-commerce platform data (based on features like color, style, etc.) to the user’s wardrobe items (stored in the wardrobe database). **Recommendation Engine:** The recommendation engine generates outfit recommendations based on the results produced by the KNN algorithm. It factors in the user’s wardrobe, occasion details, and potentially additional information like user preferences or current fashion trends. **User Interface:** The recommended outfits are presented to the user through a user interface. The user interface might allow users to browse the recommendations, select outfits, and potentially purchase items through the integrated e-commerce platforms.

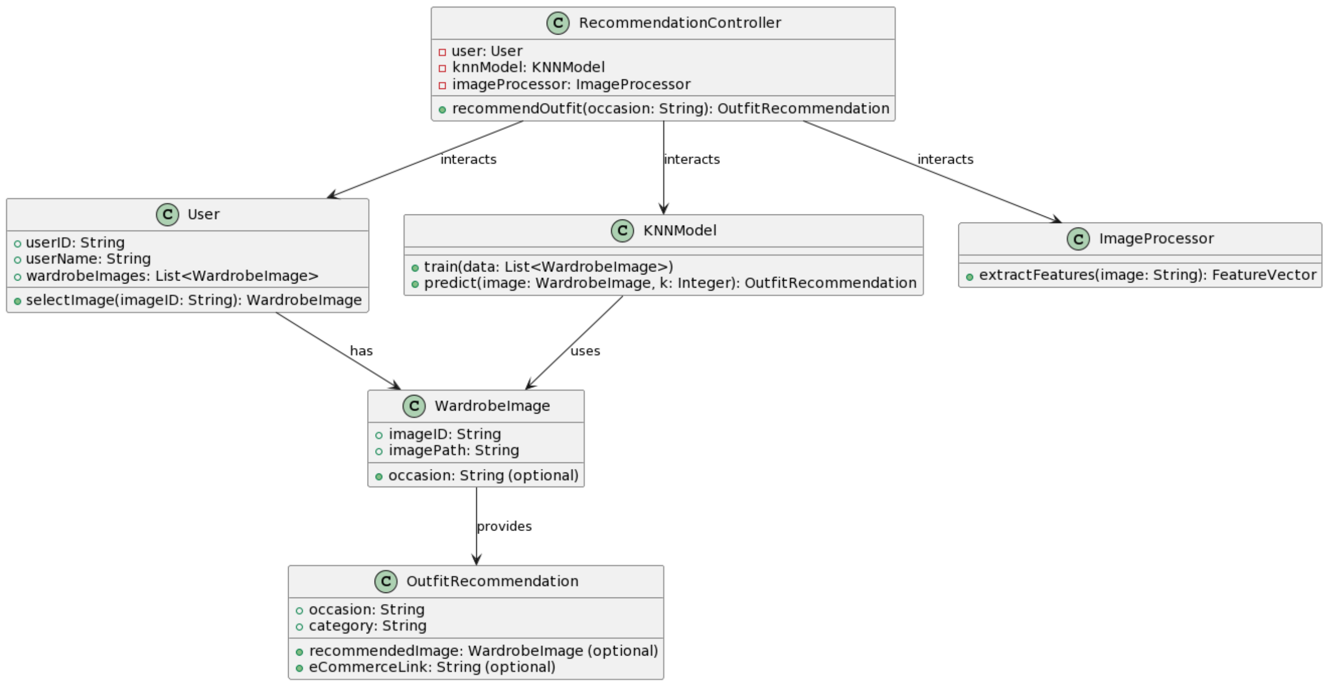

Fig 3.3 Class Diagram of Recommendation Controller

In the **Fig 3.3**, the class diagram captures the design for an outfit recommendation system that uses a K-Nearest Neighbours (KNN) model to suggest clothing based on occasion and wardrobe images. Overall, the cloud recommendation system depicted in the block diagram facilitates a personalized outfit recommendation experience by leveraging user wardrobe data, occasion details, and a KNN algorithm to generate outfit suggestions. Here's a breakdown of the classes and their interactions:

Classes

User: Represents the system's user who interacts with the application. It has attributes to store user ID, username, and a list of wardrobe images. It likely also has methods for logging in, uploading images, and interacting with the recommendation controller. **Wardrobe Image:** Represents an image uploaded by the user. It has attributes for image ID, path to the image file, and an optional occasion attribute if the user specifies it during upload. **Outfit-Recommendation:** Represents the recommendation provided by the system in response to a user's query. It includes the occasion, recommended category (e.g., casual, formal), an optional Wardrobe Image object if a suitable outfit is found in the wardrobe, and an optional e-commerce link if no matching outfit is found. **KNN Model:** Represents the K-Nearest Neighbours model used for recommendations. It has methods to train the model on a set of wardrobe images and predict the most suitable outfit category for a new image based on its nearest neighbours in the training data. **Image Processor (interface):** Represents an interface for processing images. This allows flexibility in using different image processing libraries for feature extraction. **Recommendation Controller:** This class orchestrates the recommendation process. It likely has a reference to the logged-in user and interacts with other classes to handle recommendations.

Relationships:
User owns Wardrobe Image: A user can have a collection of wardrobe images. **User interacts with Recommendation Controller:** The user interacts with the system through the recommendation controller to get outfit recommendations. **KNNModel trains on Wardrobe Image:** The KNN model is trained on a collection of labelled wardrobe images, where each image has an associated occasion label. **KNNModel predicts on Wardrobe Image:** The KNN model takes a new wardrobe image as input and predicts its category based on the trained model. **Recommendation Controller uses Image Processor:** The recommendation controller uses an image processor to extract features from the user's selected image. **Recommendation Controller uses KNNModel:** The recommendation controller uses the KNN model to get a category prediction for the user's image. **Outfit Recommendation can contain Wardrobe Image:** The recommended outfit might include an image from the user's wardrobe if a suitable match is found.

How it Works:
User Input: The user enters the occasion (e.g., birthday, wedding) and might select an image from their wardrobe. **Image Processing:** The selected image is processed by the image processor to extract relevant features. **KNN Prediction:** The KNN model predicts the outfit category (e.g., casual, formal) for the user's image based on the extracted features and its knowledge from the training data. **Recommendation Generation:** The recommendation controller creates an Outfit Recommendation object. **Matching Outfit Search:** The system searches the user's wardrobe images for items that match the occasion and predicted category. **Recommendation Output:** If a matching outfit is found, the Outfit Recommendation object includes the recommended image from the wardrobe. If no matching outfit is found, the system might generate an e-commerce link for similar outfits based on the predicted category. **User Interface:** The recommendation controller likely interacts with the user interface to display the recommended outfit details (category, image, or shopping link).

In inference, the quality of the outfit recommendations would likely depend on the quality of the data in the system's databases, particularly the comprehensiveness and detail of the user wardrobe data. The more detail and accurate the data, the better the recommendations are likely to be. The accuracy of the KNN algorithm would also be a significant factor. In order to make effective recommendations, the algorithm must be able to accurately identify similar clothing items based on the image data. It is not clear from the diagram whether the system offers any mechanism for users to provide feedback on the recommendations they receive. User feedback could be a valuable tool for improving the accuracy of the recommendation engine over time. The system could be improved by allowing users to indicate whether they found the recommendations helpful or not.

Result

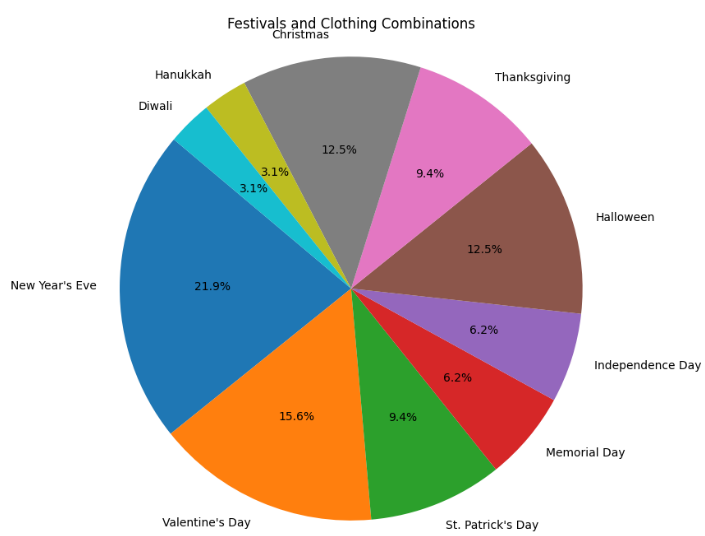

Fig 4.1 Pie Chart based on Festivals and Clothing Combinations

From the **Fig 4.1**, the pie chart depicting the percentage of people who wear different colored clothing during various festivals and holidays. This pie chart provides a general look at color trends in festival clothing choices, suggesting potential cultural associations and variations across holidays Here's a breakdown of the information it shows:

Festivals: The pie chart is divided into sections representing different festivals and holidays, including Christmas, Hanukkah, Thanksgiving, Diwali, Halloween, New Year's Eve, Independence Day, Memorial Day, Valentine's Day, and St. Patrick's Day. **Colors:** Each section of the pie chart is further divided into colored slices representing the percentages of people who wear specific colors during that particular festival. The colors include red, green (two shades), blue, orange, and purple (not labeled but appears in some sections). **Percentages:** The size of each colored slice indicates the percentage of people who wear that color during the corresponding festival. The percentages are not directly labeled but appear to range from around 3% to over 20%.

Observations:

Christmas and New Year's Eve seem to have the highest percentage of people wearing red (around 22% each), followed by Independence Day (around 15%). This likely reflects a cultural association of red with these holidays. Green appears to be a popular color choice across many festivals, though the shades differ. Lighter green might be more prevalent for St. Patrick's Day (around 12.5%), while darker green might be more common for Thanksgiving (around 9.4%). Blue and orange seem to be worn less frequently across most festivals, though there might be specific occasions where they are more prominent (e.g., blue for Hanukkah?).

Limitations:

The data source and sample size used to generate these percentages are not provided. The pie chart doesn't reveal other clothing aspects like style or type (e.g., dresses vs. jeans).

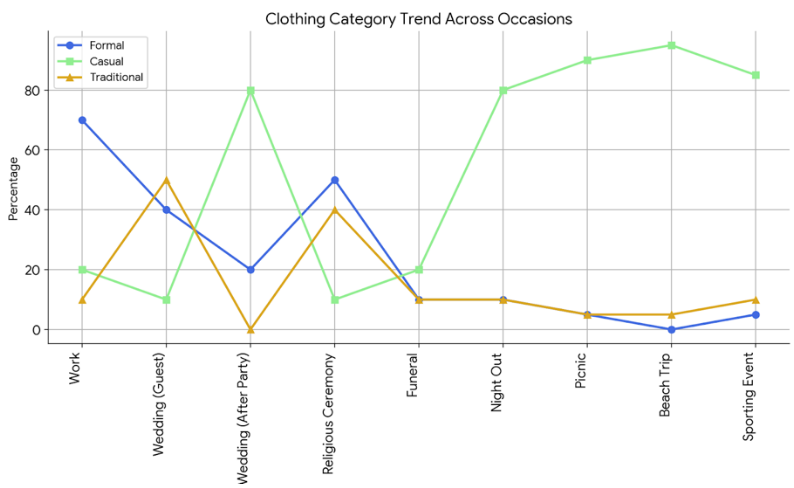

Fig 4.2 Line graph for varied Clothing Category Trend Across Occasions

The line graph depicts trends in clothing categories across various occasions. The y-axis represents the percentage, likely ranging from 0 to 80%. The x-axis lists specific occasions including work, weddings (guest and after party), religious ceremonies, funerals, night outs, picnics, beach trips, sporting events. From the data presented, formal wear appears to be the most popular clothing choice for work occasions, with a percentage hovering around 80%. Formal attire is also prevalent at weddings (guest), reaching close to 80% on the graph. There seems to be a more balanced preference for casual and traditional wear across occasions like religious ceremonies, funerals, picnics, beach trips and sporting events. None of these categories reach a dominant level of 80% on the graph, suggesting a mix of styles worn depending on the occasion. Interestingly, night outs appear to favour casual wear over formal wear, with the casual line reaching nearly 80% on the graph. Finally, traditional wear seems to be the preferred clothing choice for weddings (after party), with a percentage close to 60%.

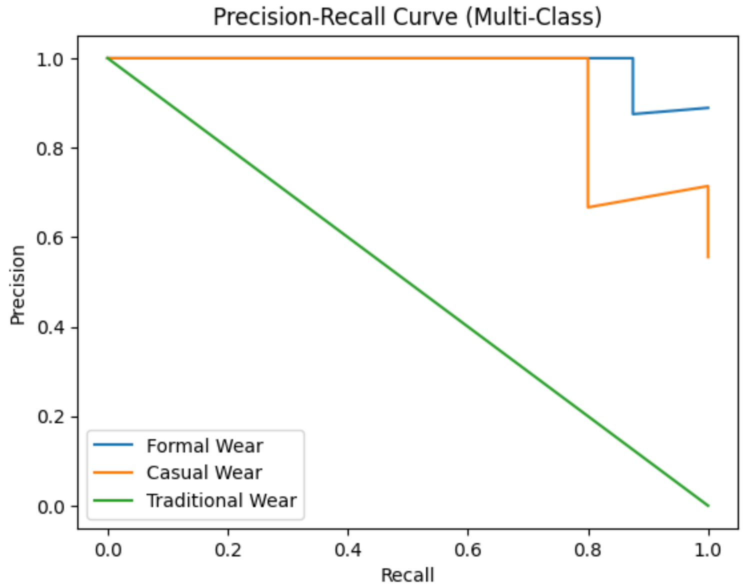

Fig 4.3 Precision Recall Curve for multi-class apparels

In the **Fig 4.3**, the curve is a precision-recall curve for a multi-class classification task. The curve depicts the trade-off between precision and recall for different classification thresholds. In the context of this graph, precision refers to the proportion of clothing items that are correctly classified as a specific type (e.g., formal wear), and recall refers to the proportion of clothing items of a specific type that are actually identified by the model. Here's a more detailed explanation of the graph:

X-axis (Recall): This axis represents the recall for each clothing category. Recall indicates how effectively the model identifies all clothes belonging to a particular category. A higher recall signifies that the model captures most of the clothing items within that category. **Y-axis (Precision):** This axis represents the precision for each clothing category. Precision reflects the accuracy of the model's classifications. A higher precision indicates that a higher proportion of the items the model identifies as a particular category are truly part of that category. **Curve:** The curve illustrates the relationship between precision and recall for different classification thresholds. As the threshold for classifying an item into a specific category becomes more stringent (meaning the model requires a higher degree of certainty for classification), the precision generally increases, but the recall typically decreases. This is because the model becomes more conservative in its classifications, potentially missing some relevant items (lower recall) but achieving higher accuracy for the items it does classify (higher precision). **Multi-class:** The curve includes multiple lines, one for each clothing category (formal wear, casual wear, and traditional wear). This allows us to compare the model's performance in classifying different garment types.

It appears that casual wear achieves the highest precision and recall compared to formal wear and traditional wear. This suggests that the model is more successful at accurately identifying casual clothing items. There could be several reasons for this:

Data Bias: The training data used for the model might have contained a larger portion of casual wear items compared to formal or traditional wear. This can lead to the model being better at recognizing patterns associated with casual wear. **Visual Characteristics:** Casual wear might exhibit more diverse visual features like colors, patterns, and styles. This variety can provide the model with more information to distinguish casual wear from other categories. **Cultural Diversity:** Traditional wear can encompass various styles depending on the specific culture or region. The model might struggle to capture the intricacies of different traditional clothing styles, especially if the training data lacked sufficient examples.

Overall, the precision-recall curve provides valuable insights into the performance of a multi-class clothing classification model. While the model seems to perform well for casual wear, there's potential for improvement in recognizing formal and traditional wear categories with more balanced training data and a model designed to handle the unique characteristics of such clothing.

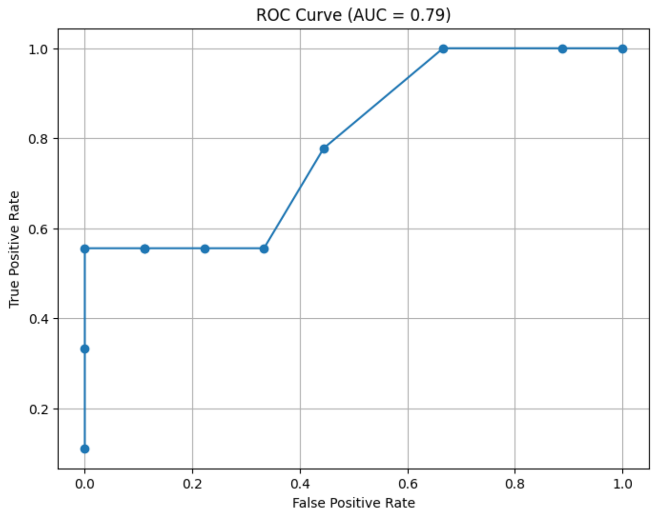

Fig 4.4 ROC Curve for Working of Model Performance

The ROC curve plots the True Positive Rate (TPR) on the y-axis and the False Positive Rate (FPR) on the x-axis. It depicts the model's performance at classifying samples across all possible classification thresholds.
General Trend: The ROC curve in the image starts at (0,0) and ends near (1,1), which is a positive sign. It suggests the model performs better than random guessing (diagonal line from (0,0) to (1,1)). As the threshold for classifying a sample as positive increases, the TPR (correctly classified positive instances) generally decreases, and the FPR (incorrectly classified positive instances) increases. **AUC (Area Under the Curve):** The AUC is a numerical value between 0 and 1 that summarizes the model's performance across all thresholds. A higher AUC indicates better overall performance. Unfortunately, the value isn't displayed in the image. **X-axis (False Positive Rate - FPR):** This axis represents the proportion of negative samples (incorrectly) classified as positive. In the context of your multi-class problem, it likely represents the combined FPR across all classes.
Y-axis (True Positive Rate - TPR): This axis represents the proportion of positive samples correctly classified as positive. Again, in your case, it likely represents the combined TPR across all classes. **Interpreting the curve without AUC:** While the specific AUC value (Area Under the Curve) isn't displayed in the image, here are some general observations about the curve's shape that can provide insights into the model's performance: **Overall Trend:** The curve starts at (0,0) and ends near (1,1). This is a positive sign, indicating the model performs better than random guessing (diagonal line from (0,0) to (1,1)). **Shape:** The curve leans closer to the top-left corner compared to a random guess line, suggesting a good ability to distinguish between positive and negative classes across all three categories (combined).

Limitations without AUC:
Without the specific AUC value, a more nuanced analysis is difficult. However, here are some general observations: **The curve leans closer to the top-left corner** compared to a random guess line, indicating a good ability to distinguish between positive and negative classes. **A steeper initial rise** would suggest a more significant improvement over random guessing at lower thresholds. **A smoother curve** throughout might indicate a more balanced performance across all thresholds.

This ROC curve represents the overall performance for all three clothing classes. It would be beneficial to see class-wise ROC curves to understand how well the model performs for each specific type of clothing. Since it's a multi-class problem, the provided code calculates the ROC AUC score using the "one-vs-rest" strategy. This approach treats each class separately and builds multiple binary classifiers.

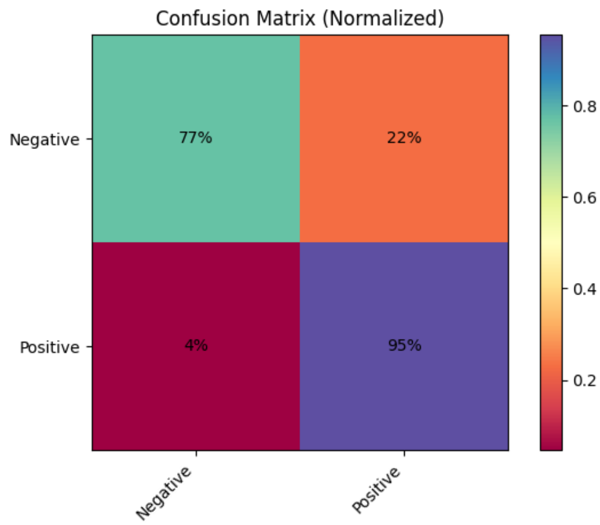

Fig 4.6 Confusion Matrix based on user's occasion

In the **Fig 4.6**, the confusion matrix in the project on recommending clothes category based on the user's occasion.

Classes: Instead of generic "positive" and "negative," your classes would be the different clothing categories you recommend, like "Formal," "Casual," "Sportswear," etc. **Cells:** The confusion matrix would show how well your model recommends the correct clothing category based on the user's occasion. **TP (True Positives):** When the model recommends a category (e.g., "Formal") that matches the user's occasion (e.g., job interview). **FP (False Positives):** When the model recommends a category (e.g., "Formal") that doesn't match the occasion (e.g., casual hangout). **FN (False Negatives):** When the model recommends a category (e.g., "Casual") that doesn't match a formal occasion. **TN (True Negatives):** When the model correctly avoids recommending an unsuitable category (e.g., not recommending "Sportswear" for a wedding).

Benefits for your project:
Evaluating Model Performance: The confusion matrix helps you identify areas for improvement. For example, high FN for "Formal" might indicate the model needs better training on formal occasions. **Understanding User Experience:** It can reveal how well your recommendations align with user expectations. High FP for "Formal" could suggest users perceive some recommendations as overly formal.
Customization:
Number of Classes: Tailor the clothing categories to your project's scope. You can have broad categories or include subcategories like "Business Casual" or "Formal Dresses." **Data Distribution:** The confusion matrix might require adjustments based on your data. If you have more formal occasions in your dataset, you might see a higher number of instances for "Formal" categories compared to "Casual." **Beyond Accuracy:** While accuracy (like the 88.88% in the example) is a good starting point, remember the confusion matrix provides a deeper understanding. Use it alongside other metrics like precision (percentage of recommended items that are relevant) and recall (percentage of relevant items recommended) to get a complete picture of your model's performance.

Discussion

The recommendation engine architecture offers a user-centric approach to outfit suggestions. Users provide details about the occasion and upload photos of their existing wardrobe. The system leverages a machine learning technique called K-Nearest Neighbors (KNN) to identify similar clothing items within the user's wardrobe based on a chosen seed image. This seed image acts as a reference point for building outfit recommendations. The engine then generates suggestions that include both outfit categories (tops, bottoms, dresses) and specific outfit combinations. User data, like style preferences, might also be considered to recommend complementary items that pair well with the existing wardrobe. Interestingly, the system seems to have a backup plan for situations where KNN doesn't yield suitable options. It might recommend links to external websites for purchasing additional items or suggest a completely different outfit based on the occasion and user data. The success of this system relies heavily on two factors: the detail and comprehensiveness of the user wardrobe data, and the accuracy of the KNN algorithm in finding similar clothing. Additionally, incorporating user feedback on the recommendations could be crucial for continuous improvement. Overall, this architecture utilizes machine learning to create a personalized recommendation experience, empowering users to leverage their existing wardrobe while potentially suggesting complementary purchases. This approach to fashion recommendations can be both sustainable and cost-effective for users.

Conclusion

In conclusion, the research presents an innovative solution, Smart Wardrobe, that revolutionizes the way users manage their clothing selections and outfit creations through the integration of image recognition and machine learning technologies. By enabling users to upload photos of their existing garments, the application employs K-Nearest Neighbors (KNN) algorithm to analyze visual features such as color, style, and pattern. This analysis facilitates the recommendation of outfit combinations tailored to specific occasions, enhancing user convenience and style. Moreover, Smart Wardrobe seamlessly integrates e-commerce functionality, providing users with recommendations for missing wardrobe components and links to relevant online stores. This holistic approach transforms Smart Wardrobe into a personalized fashion assistant, empowering users to optimize wardrobe utilization and streamline the clothing selection process. Ultimately, Smart Wardrobe not only offers outfit suggestions but also serves as a comprehensive tool that enhances user wardrobe management and fosters a more efficient and stylish lifestyle.

References

[1] AŞIROĞLU, B., Atalay, M. I., Balkaya, A., TÜZÜNKAN, E., Dağtekin, M., & ENSARİ, T. (2019, October). Smart clothing recommendation system with deep learning. In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1-4). IEEE.

[2] Liang, J. (2020, September). Image classification based on RESNET. In Journal of Physics: Conference Series (Vol. 1634, No. 1, p. 012110). IOP Publishing.

[3] Guan, C., Qin, S., Ling, W., & Ding, G. (2016). Apparel recommendation system evolution: an empirical review. International Journal of Clothing Science and Technology, 28(6), 854-879.

[4] Zhou, Z., Liu, M., Deng, W., Wang, Y., & Zhu, Z. (2022). Clothing image classification algorithm based on convolutional neural network and optimized regularized extreme learning machine. *Textile Research Journal*, *92*(23-24), 5106-5124.

[5] Shin, S. Y., Jo, G., & Wang, G. (2023). A novel method for fashion clothing image classification based on deep learning. Journal of Information and Communication Technology, 22(1), 127-148.

[6] Murtaza, M., Sharif, M., Yasmin, M., & Kadry, S. (2022). A novel approach of boundary preservative apparel detection and classification of fashion images using deep learning. *Mathematical Methods in the Applied Sciences*.

[7] Liang, J. (2020, September). Image classification based on RESNET. In Journal of Physics: Conference Series (Vol. 1634, No. 1, p. 012110). IOP Publishing.

[8] Shakya, S., Shrestha, B., Thapa, S., Chauhan, A., & Adhikari, S. (2021). Clothes Identification Using Inception ResNet V2 and MobileNet V2. *Available at SSRN 3949190*.

[9] Seo, Y., & Shin, K. S. (2019). Hierarchical convolutional neural networks for fashion image classification. *Expert systems with applications*, *116*, 328-339.

[10] Rohrmanstorfer, S., Komarov, M., & Mödritscher, F. (2021). Image classification for the automatic feature extraction in human worn fashion data. *Mathematics*, *9*(6), 624.

[11] Cychnerski, J., Brzeski, A., Boguszewski, A., Marmolowski, M., & Trojanowicz, M. (2017, September). Clothes detection and classification using convolutional neural networks. In *2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA)* (pp. 1-8). IEEE.

[12] Singh, M., Dalmia, S., Ranjan, R. K., & Singh, A. (2021, December). Dress Pattern Classification Using ResNet Based Convolutional Neural Networks. In *International Conference on Information Systems and Management Science* (pp. 91-103). Cham: Springer International Publishing.