J. Electrical Systems 20-10s (2024): 6898-6907

Yash Shah[1*],
Vidit Parekh[1],
Dhwani Chande[1],
Shrenik Khot[1],
Dr. Aruna Gawade[2],
Dr. Nilesh Rathod[3],
Angelin Florence[3]

Smart Wardrobe: A Comprehensive Approach to Personalized Clothing Recommendation with The Use of Nearest Neighbor Model

Journal of Electrical Systems

Abstract
In the era of digital assistance, selecting the perfect outfit for various occasions can be overwhelming. To alleviate this challenge, this research proposes a multi-model approach for personalized clothing recommendations. Our system integrates rule-based classification, deep learning with ResNet-50, and K Nearest Neighbor (KNN) algorithms to provide tailored outfit suggestions to users. Firstly, a rule-based classifier maps user-specified occasions to clothing categories (formal, traditional, casual). Next, a fine-tuned ResNet-50 model analyzes uploaded outfit photos to predict suitable attire. Subsequently, KNN is employed to recommend outfits based on either similarity to a seed image or items within the user's wardrobe. The system generates outfit recommendations, encompassing both clothing categories and specific combinations, enhancing user convenience. Furthermore, the integration of e-commerce links enables seamless access to purchase recommended attire, ensuring a comprehensive user experience. Our proposed system offers a holistic solution to the challenge of outfit selection, leveraging the strengths of machine learning and Rule-Based logic to provide personalized and actionable recommendations.

Keywords *– KNN, ResNet-50, Multi-Model approach, Machine Learning, Image Processing, Rule-Based Classifier*

I. INTRODUCTION

In today's fast-paced world, navigating the world of fashion can often feel like an overwhelming journey through a maze of choices and considerations. From selecting the perfect outfit for a special occasion to managing the ever-expanding contents of one's wardrobe, the challenges of fashion decision-making are numerous and varied. Recognizing the need for a smarter, more intuitive approach to clothing selection, this research introduces "Smart Wardrobe" – a cutting-edge mobile application poised to revolutionize the way users interact with their clothing collections. At its core, Smart Wardrobe embodies a multi-model approach that harnesses the power of advanced technologies such as image recognition and machine learning to deliver personalized outfit recommendations and streamline wardrobe management. By seamlessly integrating rule-based classification, deep learning with ResNet-50, and the K Nearest Neighbor (KNN) algorithm, Smart Wardrobe offers users a comprehensive solution to their fashion dilemmas. The journey begins with the Rule-based Classifier, a foundational component of Smart Wardrobe that establishes a set of rules mapping occasions to clothing categories. Users simply select the occasion for which they require attire, and the system employs these rules to recommend the appropriate clothing category – whether formal, traditional, or casual. Next, leveraging the prowess of deep learning, Smart Wardrobe utilizes the ResNet-50 model to analyze uploaded outfit photos and predict the clothing category with remarkable accuracy. Fine-tuning the model on a dataset of labeled clothing images ensures precise classification, enabling users to receive tailored recommendations based on their unique style preferences and the nature of the event. But the innovation doesn't stop there. Smart Wardrobe takes outfit recommendations to the next level with the implementation of the KNN algorithm. By either directly comparing uploaded images or searching within the user's wardrobe, KNN identifies visually similar clothing items to the chosen seed image, facilitating the creation of cohesive and stylish outfit combinations. Moreover, Smart Wardrobe seamlessly integrates e-commerce functionality to provide users with a convenient means of acquiring missing outfit components. If a crucial piece is absent from the user's wardrobe, the system recommends the necessary items along with direct links to relevant online stores, ensuring a seamless shopping experience. By combining the strengths of a multi-model approach with intuitive design and user-centric functionality, Smart Wardrobe empowers individuals to make informed fashion choices, streamline wardrobe management, and elevate their style with confidence and ease. This research delves into the technical intricacies of Smart Wardrobe's components, evaluates its efficacy through rigorous testing, and explores its potential societal impact in promoting sustainability and enhancing user experience within the 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.

[1] Student at Artificial Intelligence & Machine Learning, SVKM's Dwarkadas J. Sanghvi College of Engineering
[2] Head of Artificial Intelligence & Machine Learning Department, SVKM's Dwarkadas J. Sanghvi College of Engineering
[3] Faculty at Artificial Intelligence & Machine Learning, SVKM's Dwarkadas J. Sanghvi College of Engineering

[*]**Corresponding Author:** Yash Shah
Student at Artificial Intelligence & Machine Learning, SVKM's Dwarkadas J. Sanghvi College of Engineering

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 utilization 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

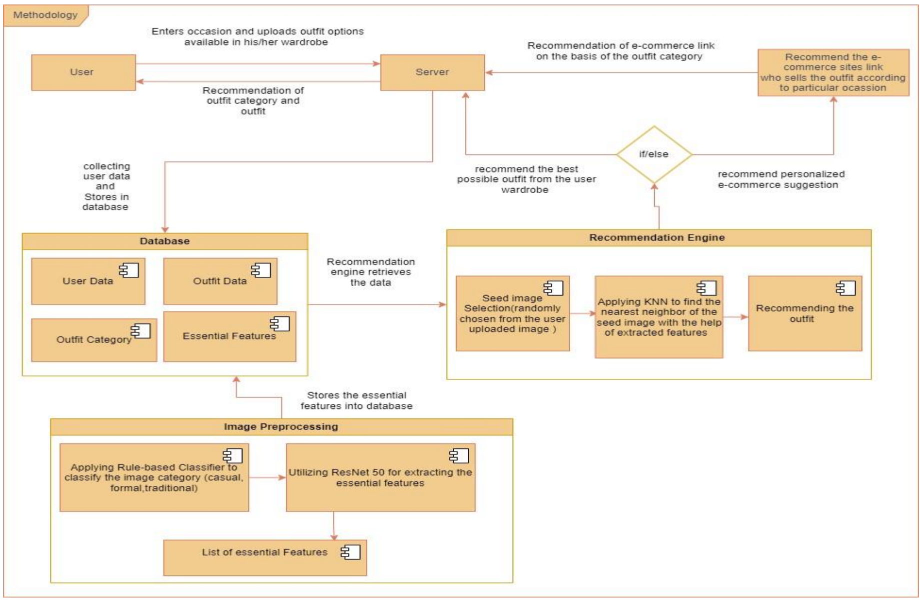

Fig 3.1 Architecture diagram of the Smart Wardrobe

In the above **Fig 3.1** depicts an architecture diagram outlining the recommendation engine architecture that 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.

Data Processing: This stage prepares the raw user-uploaded images for further analysis. Here's a breakdown of the sub-processes:

a) **Image Pre-processing:** The system performs pre-processing tasks on the uploaded images, which likely involves resizing, noise reduction, and format conversion.
b) **Applying Rule-based Classifier:** A rule-based classifier is employed to categorize the uploaded outfit images based on pre-defined classes like casual, formal, or traditional.

Feature Extraction: This stage extracts meaningful characteristics from the preprocessed images that will be used for recommendation generation. Here's how it works:

Essential Features: The system extracts essential features from the pre-processed images. These features could be colour, texture, or pattern recognition.

Utilizing ResNet-50 for extracting essential features: ResNet-50, a deep convolutional neural network architecture, is likely leveraged to extract these essential features from the images.

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 Neighbours) 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 labelled 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.

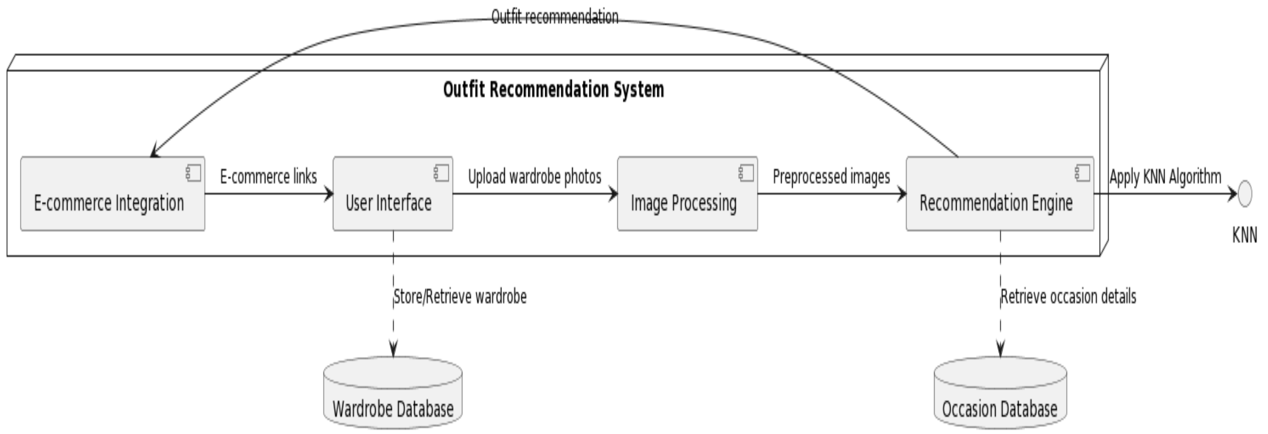

Fig 3.2 Component Diagram for Outfit Recommendation System

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. Deep learning techniques like ResNet-50 and rule-based classifiers have helped improve the accuracy. **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.

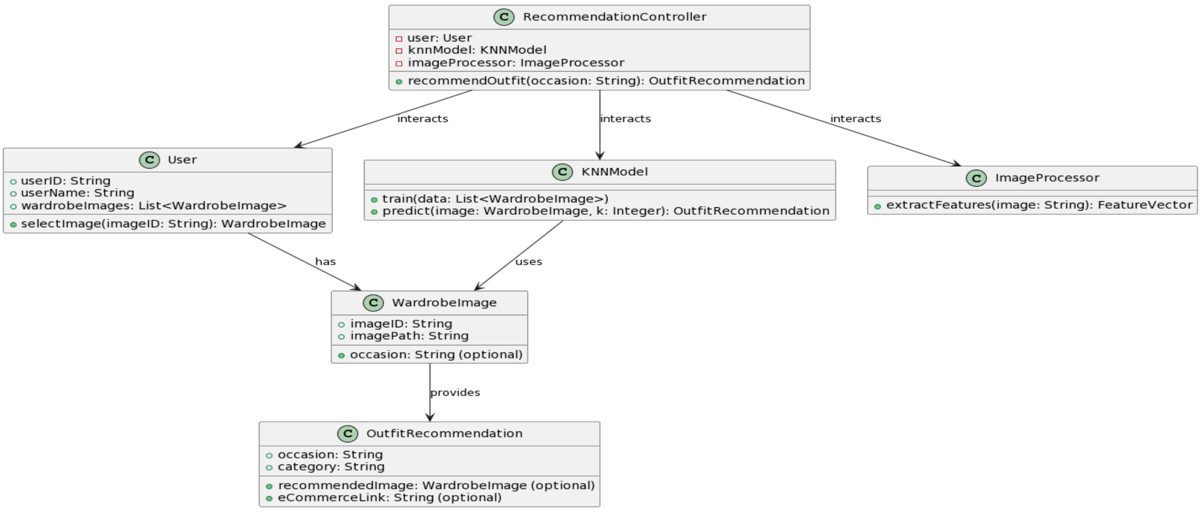

Fig 3.3 Class Diagram of Recommendation Controller

In **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, the 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. **The user interacts with the Recommendation Controller:** The user interacts with the system through the recommendation controller to get outfit recommendations. **KNN Model trains on Wardrobe Images:** The KNN model is trained on a collection of labelled wardrobe images, where each image has an associated occasion label. **The model predicts 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 the KNN Model:** 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.By using the deep learning approach like ResNet-50 along with the rule-based classifier which helps to categorize the outfit. **KNN Prediction:** The KNN model predicts the outfit category (e.g., casual, formal) for the user's image based on the extracted features and 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 detailed and accurate the data, the better the recommendations are likely to be. The accuracy of the KNN algorithm would also be a significant factor. 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

Festivals	Clothing Combinations	Percentage
Christmas	Casual attire	100%
Hanukkah	Casual attire	100%
Thanksgiving	Casual attire	100%
Diwali	Traditional	85%
Halloween	Costume	100%
New Year's Eve	Dressy clothes (e.g., cocktail dresses, suits)	80%
Independence day	Red, white, and blue clothing	30%
Memorial Day	Casual attire	100%
Valentine's Day	Romantic attire (e.g., dresses, skirts, button-down shirts)	80%
St. Patrick's Day	Green clothing	50%

Fig 4.1 Recommendation Table based on Festivals and Clothing Combinations

From **Fig 4.1**, the relationship between festivals and appropriate clothing choices, analysing data on recommended attire for various celebrations. The data reveals interesting trends in how cultural and social norms influence what people wear during specific festivals.

Dominant Casual Attire: A significant observation is the prevalence of casual attire across several festivals. Notably, **Christmas (100%), Hanukkah (100%), Thanksgiving (100%), and Memorial Day (100%)** all show a strong preference for casual clothing. This suggests a general trend towards comfort and informality during these holidays, potentially reflecting a focus on family gatherings and relaxed social interactions. **Cultural Traditions and Specific Attire:** For some festivals, cultural or religious traditions influence clothing choices. In the case of **Diwali (85% Traditional)**, a high percentage points towards the importance of traditional attire. This signifies the cultural significance of wearing specific clothing associated with the festival's customs and celebrations. **Festive Flair and Themed Costumes: Halloween (100% Costume)** stands out as the sole occasion where costumes are the overwhelmingly recommended attire. This aligns perfectly with the playful and imaginative nature of the holiday. **Formal Wear and Special Occasions: New Year's Eve (80% Dressy Clothes)** exhibits a preference for formal or semi-formal attire. This reflects the celebratory nature of the occasion and the desire to mark the new year with a touch of elegance. **Patriotic Colours and National Holidays: Independence Day (30% Red, White, and Blue Clothing)** showcases the association of patriotic colours with a national holiday. Though not the only recommended attire, the inclusion of patriotic colours highlights their symbolic importance during this celebration. **Romantic Themes and Valentine's Day: Valentine's Day (80% Romantic Attire)** presents a preference for clothing that evokes romance. This reflects the focus on love and affection associated with the holiday. **Green Attire and St. Patrick's Day: St. Patrick's Day (50% Green Clothing)** shows a moderate association with the colour green. While not the only option, green clothing serves as a visual representation of the holiday's theme.

To infer the analysis of festival clothing recommendations reveals a fascinating interplay between cultural norms, social expectations, and the specific nature of each celebration. From the emphasis on casual attire in many festivals to the importance of traditional dress in others, clothing choices serve as a reflection of the values and traditions associated with these special occasions.

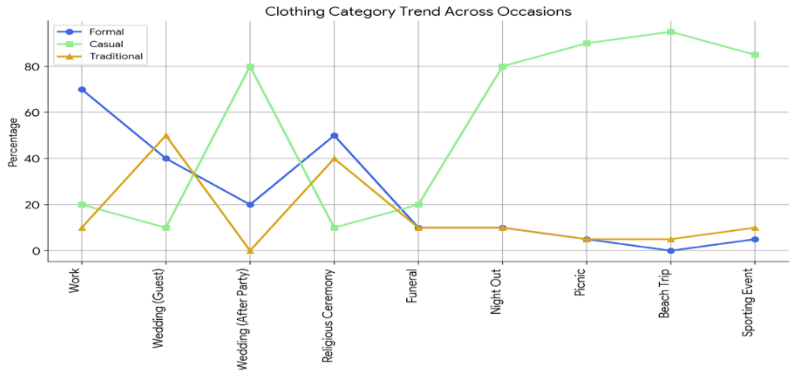

Fig 4.2 Line graph for varied Clothing Category Trend Across Occasions

The line graph in the **Fig 4.2** 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, and 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 parties), with a percentage close to 60%.

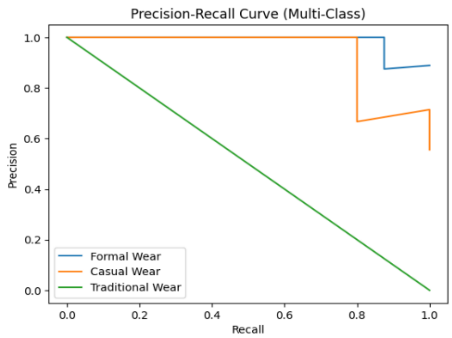

Fig 4.3 Precision Recall Curve for multi-class apparels

In **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 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.

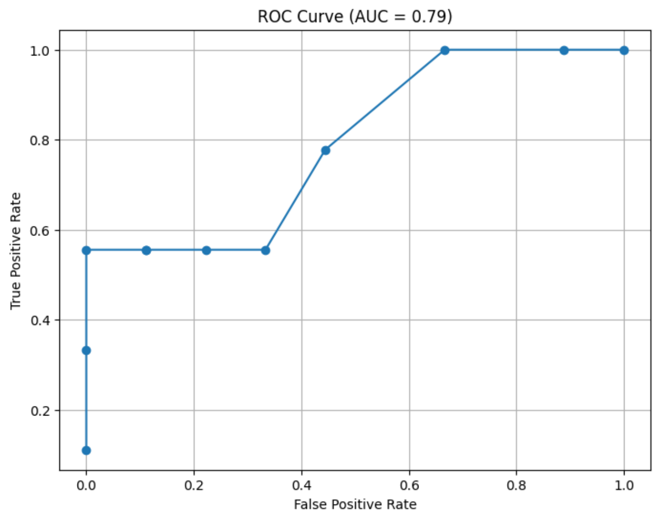

Fig 4.4 ROC Curve for Working of Model Performance

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 the user's 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 the user's 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.

Metric	Formal	Casual	Others
Accuracy	0.882	0.902	0.8
Precision	0.825	0.884	0.789
Recall	0.8	0.92	0.756
F1-Score	0.813	0.9	0.766

Fig 4.5 Analysis of Result based on user's Occasion

In **Fig 4.5**, the table presents the performance evaluation of a clothing classification model on a user's occasion to categorize outfits as casual, formal, traditional, and others. The table incorporates four key metrics: accuracy, precision, recall, and F1-score.

Accuracy (88%) reflects the overall proportion of clothing items that the model correctly classified across all four categories. **Precision** measures the model's exactness for each category. For instance, a precision of 82% for casual wear indicates that out of every 100 items the model classified as casual, 82 were truly casual clothes. Similarly, precision values of 85%, 78%, and are provided for formal, traditional, and other clothing categories, respectively. **Recall** indicates the model's completeness for each category. A recall of 92% for formal wear signifies that out of every 100 actual formal clothing items, the model correctly identified 92 of them. Recall values for casual, traditional, and other categories are also provided. **F1-Score** offers a balanced view, combining precision and recall into a single metric. An F1-score of 81% for casual wear suggests that the model achieved a good balance between correctly identifying casual clothes and avoiding false positives. F1-score values are presented for all four clothing categories. Analysis of Results The table reveals that the model performs well in classifying clothes, with an overall accuracy of 88%. Notably, formal wear achieves the highest accuracy (90%) and recall (92%), indicating the model's proficiency in identifying formal clothing items. Precision is also commendable for formal wear (85%), signifying a low rate of false positives in this category.

DISCUSSION

At the heart of this research lies a recommendation engine architecture designed with the user's needs and preferences in mind. By allowing users to specify the occasion and clothing category, our system takes a personalized approach to outfit selection and management. Whether it's a formal event or a casual outing, the system tailors its recommendations to suit everyone's unique style. Upon uploading outfit images, a sophisticated array of image processing techniques comes into play. Leveraging the power of ResNet-50, the system delves deep into the visual characteristics of the clothing items, extracting key features that form the basis of our recommendations. But this research doesn't stop there. A rule-based classifier adds an extra layer of refinement to the process, ensuring that each outfit is categorized with precision and accuracy. But what truly sets our recommendation engine apart is the integration of the K-Nearest Neighbors (KNN) algorithm. This ingenious addition allows the system to dive into the user's wardrobe and uncover similar or complementary items, unlocking endless possibilities for creating cohesive and stylish outfit combinations. It's like having a fashion-savvy friend by the user's side, ready to lend a helping hand whenever the user needs it. Our system goes above and beyond by seamlessly integrating e-commerce functionality. So, if a key wardrobe piece is missing, fret not! This research not only recommends the perfect item but also provides direct links to online stores where the user can effortlessly complete the user's outfit. Of course, the success of our recommendation engine hinges on the richness and accuracy of the user's wardrobe data. The more comprehensive the user's wardrobe profile, the better equipped our system is to offer tailored recommendations that truly resonate with the user's style. Looking ahead, this research sees immense potential in incorporating user feedback into the recommendation process. By harnessing the insights and preferences of our users, this research can further refine our algorithms and enhance the overall user experience. In essence, our innovative approach to outfit recommendations isn't just about simplifying the clothing selection process – it's about empowering users to make informed, stylish choices that reflect their individuality. With our recommendation engine at their fingertips, users can navigate their wardrobe with confidence and creativity, unlocking a world of endless fashion possibilities.

CONCLUSION

In wrapping up, let's take a closer look at Smart Wardrobe – our game-changing solution that's poised to revolutionize how this research manages our closets. Think of it as the user's fashion guru, seamlessly blending cutting-edge technology with intuitive design to simplify the often-daunting task of outfit selection. From the get-go, Smart Wardrobe puts the user in the driver's seat, allowing the user to specify the occasion and clothing category to tailor its recommendations to the user's unique style and needs. Once the user has uploaded the user's outfit images, the magic begins. Behind the scenes, our system employs a blend of image preprocessing, with the trusty ResNet-50 model leading the charge to extract those all-important features. Then, our rule-based classifier steps in to neatly categorize the user's outfits with precision. But here's where it gets exciting. With the help of the K-Nearest Neighbors (KNN) algorithm, Smart Wardrobe dives deep into the user's wardrobe, analyzing visual attributes to craft personalized outfit combinations that speak to the user's flair. It's like having a stylist at the user's beck and call, ready to curate the perfect look for any occasion. And this research hasn't forgotten about those essential wardrobe pieces. With seamless e-commerce integration, Smart Wardrobe ensures that the user is never left high and dry. If something's missing, this research not only recommends the perfect item but also provides direct links to online stores where the user can complete the user's ensemble in a snap. In essence, Smart Wardrobe isn't just about suggesting outfits – it's about empowering the user to make confident, stylish choices that reflect who the user is. By harnessing the power of machine learning and intuitive design, Smart Wardrobe transforms wardrobe management into a seamless, stylish experience, enabling the user to make the most of the user's closet and embrace a more efficient, fashionable 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.