Vanilla
Personalized Outfit Recommendation System

Arwa Basabreen[1], Ebtesam Kaid[2], Reem Almutairi[3], Qamar Alashrafi[4], Riham Bichara[5]
[1]Email: abasabreen@kau.edu.sa
[2]Email: eaqaid@stu.kau.edu.sa
[3]Email: rawadhalmutairi@stu.kau.edu.sa
[4]Email: Qabdulazizalashrafi@stu.kau.edu.sa
[5]Email: Rmahamat@stu.kau.edu.sa

***Abstract*—The online fashion shopping experience has long faced challenges in delivering personalized and accurate recommendations to users. This paper introduces an advanced fashion recommendation system designed to transform how users discover clothing items that align with their unique preferences and styles. The system leverages ResNet50 for deep image feature extraction, the Similarity Model for identifying visual relationships between items, and K-Means clustering for organizing these items into meaningful groups. ResNet50 extracts intricate visual features from clothing images, enabling precise analysis of attributes like texture, color, and style. These features are then processed by the Similarity Model to calculate and rank the visual closeness of items, ensuring that related products are identified efficiently. Finally, K-Means clustering categorizes the items into relevant clusters, ensuring that recommendations are efficient, relevant, and visually aligned with user preferences. The system implementation includes a user-friendly interface, ensuring a seamless and enjoyable experience. The results demonstrate the effectiveness of our system in providing tailored recommendations, empowering users to explore and discover clothing items that resonate with their unique style preferences. This paper presents a significant advancement in fashion recommendation systems, offering a promising solution to enhance the online fashion shopping experience.**

***Index Terms*—Fashion Recommendation System, ResNet50, K-Means Clustering, Deep learning.**

I. Introduction

This paper will discuss an Outfit Recommender System using models for image classification, segmentation, and detection. Moreover, similarity metrics were used to perform recommendations satisfactorily.

This project mainly aims to save time by providing accurate predictions that better customer options. Additionally, this can reduce the number of returns, which benefits both the consumer and the manufacturer. This project can also help people with eyesight problems needing external assistance.

The research in this field is very sparse currently, and although some datasets exist, we needed extensive research to carry this project successfully.

We first prepared the data by preprocessing and augmenting it to be able to test on a variety of models. To be able to build a successful recommendation system, we first leverage a pre-trained Convolutional Neural Network (ResNet50) that we retrained on our dataset to make it more fit for our field. We extract the embeddings of the images in the dataset to be able to find the similarity of images using the similarity architecture: Siamese Network. This is to place similar items closer and dissimilar items further. Using the results of the similarity architecture, we use a K-Means Clustering model for the recommendation model to recommend the closest fitting items to the input item to have a complete outfit.

We also performed segmentation and detection by retraining two known models DeepLabV3 and Yolo to be able to add the bounding boxes and pixel annotations.

We also designed a website using Flask that showcases the classification of the input image, the confidence score, the recommended outfit, the image segmentation, and the image detection. To successfully use it, we deploy it and apply a database to store all relative files.

II. Literature Review

Fashion recommendation systems have gained a lot of attention in recent years due to the demand for more tailored and user-friendly purchasing experiences. These technologies use breakthroughs in artificial intelligence and machine learning to deliver customized outfit suggestions based on individual tastes and styles. Fashion recommendation systems seek to improve customer pleasure and engagement.

This literature review explores previous papers that influenced the development of these systems. It investigates different techniques and approaches, emphasizing their contributions to increasing customization, compatibility, and scalability in fashion recommendations. Understanding these achievements lays the groundwork for generating effective solutions in the field.

Fashion recommendation systems have gained significant attention in recent years, driven by the need for personalized and user-friendly shopping experiences. These systems leverage advancements in artificial intelligence and machine learning to provide tailored outfit suggestions that align with individual preferences and styles. By combining user data with innovative algorithms, fashion recommendation systems aim to enhance customer satisfaction and improve engagement in online retail.

Notable studies such as, ”Diffusion Models for Generative Outfit Recommendation” [1] and ”POG: Personalized Outfit

Generation for Fashion Recommendation at Alibaba iFashion" [2], offer innovative approaches to advancing fashion recommendation systems. Xu et al. introduce Generative Outfit Recommendation (GOR) with their model, DiFashion, which uses diffusion processes to iteratively refine noisy inputs into realistic fashion images, guided by category prompts, mutual compatibility, and user history [1]. Chen et al., on the other hand, propose Personalized Outfit Generation (POG), which employs a Transformer-based encoder-decoder architecture to generate personalized and compatible outfits based on user behavior and item interactions [2].

The study "K-Means and Morphology-Based Feature Element Extraction Technique for Clothing Patterns and Lines", introduces a hybrid method for extracting patterns and lines from clothing images. The authors utilized K-means clustering to segment the images into regions based on RGB pixel values, which allowed for clear identification of patterns. After segmentation, morphological operations like erosion and dilation were applied to enhance and detect clothing lines by removing noise and refining boundaries. The method was tested on a dataset consisting of 500 clothing images with varying designs and patterns. The proposed approach achieved a classification accuracy of 94.7%, highlighting its efficiency in pattern detection and line extraction. Additionally, the system reduced computational time compared to traditional techniques, making it ideal for applications like garment quality control and automated image analysis in the fashion industry. [3]

The study "Fashion Recommendation System", presents a machine-learning-based recommendation model designed to provide personalized clothing suggestions. The authors combined collaborative filtering and content-based techniques to analyze 5000 clothing items and user behavior, including purchase history and reviews. The system uses machine learning algorithms to predict user preferences and recommend relevant clothing options, improving e-commerce user experience. The results demonstrated a recommendation accuracy of 89%, showing the system's effectiveness in delivering accurate suggestions. The study also emphasizes the scalability of the proposed method, making it suitable for integration into large-scale online fashion platforms to enhance customer satisfaction and drive sales.[4]

These studies highlight the transformative potential of advanced techniques in the fashion industry, particularly through machine learning and image processing. By enhancing feature extraction, improving recommendation accuracy, and enabling automated analysis, they provide innovative solutions for addressing complex challenges in garment identification and personalized recommendations. These approaches balance creativity, practicality, and scalability, offering tools that improve efficiency, user experience, and real-world applications in fashion systems. Together, they demonstrate how AI and machine learning can drive advancements in fashion technology, meeting industry needs while delivering enhanced outcomes for both businesses and consumers.

III. Dataset Analysis

For this project, a new dataset was created from a previously existing dataset, *H&M Personalized Fashion Recommendations* [5]. The original dataset, sourced from Kaggle, contains information related to customer shopping behavior and clothing items sold by H&M. It includes the following data:

- **Type of data:** Images (of clothing items) and transactional data (e.g., customer IDs, product IDs, purchase history).
- **Number of samples:** The image dataset represents approximately 1 million rows of transactional data and over 20,000 clothing items.
- **Characteristics:** The dataset provides various features, such as product category, product name, customer purchase history, and images of the items. Different clothing types categorize the images, and the dataset offers a mix of labeled and unlabeled data, ideal for training models in a real-world recommendation system.

Due to limited GPU and time constraints, the dataset was reduced to six classes of clothing items:

- Trousers
- Dress
- Sweater
- T-shirt
- Top
- Blouse

Each class contains 1,000 images, resulting in a total of 6,000 images in the final dataset. Below are some sample images from the dataset.

Fig. 1. classes: Trousers, Dress, and Top.

IV. Image Preprocessing and Augmentation

Some image preprocessing and augmentation were needed so the model could process the pictures smoothly.

1- Resizing images to 224x224 pixels as the model that was used, ResNet50, expects this type specifically.

2- To increase the size of the image dataset, several types of augmentation were applied (to the training dataset only):

  - Rotation-
  - Width/Height shift-
  - Shear-
  - Zoom-
  - Horizontal Flip-

3- All categorical labels were numerically encoded as required for model classifications.

4- Images are normalized into the range [0,1] to help with faster convergence and ensure that input is all on a similar scale.

5- The batch size of the training and validation dataset was chosen to be 32.

V. Experimental Setup

- **Hardware**
  - NVIDIA GPUs were used for model training.
  - Google Drive was used to store models and embeddings.
- **Software**
  - Python 3.x
  - Deep Learning Framework: TensorFlow, Keras
  - Libraries: TensorFlow 2.x, Keras, Scikit-learn, Matplotlib, Seaborn, NumPy, and Pandas
- **Hosting and Deployment as Web App** The project is deployed as a separated frontend and backend projects as a GitHub repository hosted on Netlify and Amazon EC2 Instance.

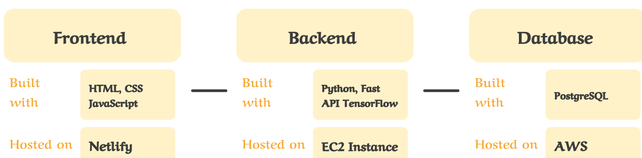

Fig. 2. Vanilla Project Architectur

Hyperparameters

- Learning rate: 0.0001 (in the optimizer)
- Batch size: 32
- Epochs: 5 (initial training), then 10 (fine-tuning)
- Regularization: L2 with strength 0.01 applied to dense layers
- Dropout: 0.5 after each dense layer to prevent overfitting

Training Details

- Training-validation split: Automatically done by data generators.
- Optimization: Adam optimizer with a learning rate of 0.0001 for faster convergence.
- Early stopping was used to prevent overfitting.
- Some layers were frozen and gradually unfrozen to improve model performance.
- Sparse categorical cross-entropy loss was employed to avoid memory-intensive one-hot vectors.
- Stratified K-Fold Cross-Validation: 5-fold cross-validation was applied, with each fold trained for 10 epochs to ensure robust evaluation.
- Siamese Network Architecture: Included shared convolutional layers for feature extraction and dense layers with L2 regularization and dropout for embedding refinement.
- Triplet Loss: Used to minimize the distance between positive pairs and maximize separation for negative pairs.

VI. Model Performance Analysis

c

MetricUsed	ResNet50 (Classification)	Yolo (Detection)	DeepLabV3 (Segmentation)	Compatibility Modeling	K-Means clustering
Accuracy	0.762	0.87	0.873	0.768	0.901
Precision	0.752	0.83	0.875	0.767	0.927
Recall	0.762	0.74	0.879	0.768	0.896
F1-Score	0.750	0.78	0.874	0.767	0.895
mAP	N/A	N/A	0.122	0.092	0.888
IoU	N/A	0.70	0.129	0.986	0.992
Dice Coefficient	N/A	N/A	0.1146	0.993	0.996

TABLE I

MODEL PERFORMANCE

ResNet50 Model Results after Retraining on Our Own Dataset

Confusion Matrix

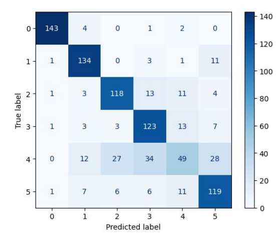

Fig. 3. Confusion matrix for ResNet50 model

Training and Validation Accuracy Graph:

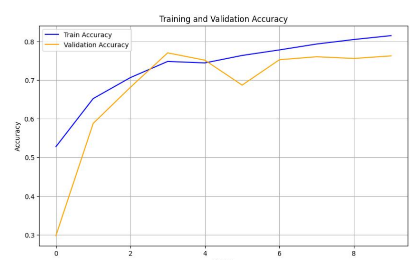

Fig. 4. Training and Validation Accuracy for ResNet50 model.

Training and Validation Loss Graph:

Fig. 5. Training and Validation Loss for ResNet50 model.

Similarity Model Results

Confusion Matrix:

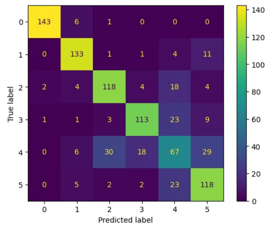

Fig. 6. Confusion matrix for Similarity model.

Training and Validation Accuracy Graph:

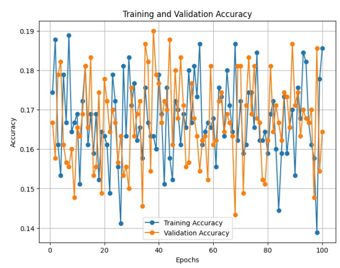

Fig. 7. Training and Validation Accuracy for Similarity model.

Training and Validation Loss Graph:

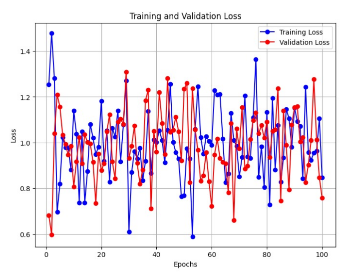

Fig. 8. Training and Validation Loss for Similarity model.

K-Means Clustering Results

Confusion Matrix:

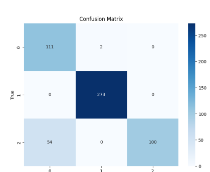

Fig. 9. Confusion matrix for K-Means clustering.

ROC Curve:

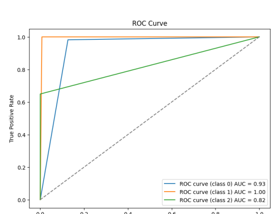

Fig. 10. ROC Curve for K-Means clustering.

VII. Discussion

The system's implementation shows clear performance differences among the models used for outfit recommendation tasks, highlighting their strengths in handling various types of input data. Vision-based models like ResNest50 excelled at extracting detailed features from outfit images. In particular, ResNest50 was able to catch small details and the relationships between elements in the photos due to its architecture, which makes use of split-attention blocks and hierarchical feature fusion. In order to enhance the system even more, a similarity search was incorporated for suggestions. This procedure was divided into multiple essential parts:

leftmargin=*

- **Embedding Extraction:** Using ResNest50's embeddings, the system mapped images into a shared feature space, which allowed for efficient pairwise comparisons.
- **Real-Time Similarity Search:** Once the embeddings were extracted, the system performed real-time queries to find items similar to user inputs, ensuring quick and relevant suggestions.
- **KNN-Based Recommendation System:** A K-Nearest Neighbors (KNN) model was trained to group similar items and offer ranked recommendations. Hyperparameters were optimized to balance accuracy and computational efficiency, while the query mechanism ensured smooth retrieval of compatible items.

Compatibility Modeling

To ensure recommendations matched user preferences for categories, colors, patterns, and occasions, compatibility between items was modeled in two steps:

- **Pair Generation:** Compatibility rules were defined based on visual and contextual factors like patterns, colors, and suitability for specific occasions. Using these rules, item pairs were generated to train the compatibility model.
- **Building the Compatibility Model:**
  - A Siamese Network was used to learn a compatibility score between item pairs. This network used shared weights for consistency in feature extraction and was trained using labeled pairs.*
  - A custom Triplet Loss function encouraged compatible items (anchor and positive) to be closer, while incompatible ones (negative) were pushed farther apart.*
  - A Triplet Network was designed to process anchor, positive, and negative samples to refine compatibility embeddings.*

The Siamese and Triplet Networks were trained using these pairs and triplets, ensuring the system could effectively distinguish between compatible and incompatible items in complex situations.

Preprocessing and augmentation also played a vital role in improving the system's robustness. Techniques like image normalization, resizing, and varied backgrounds and lighting helped the models handle diverse user inputs. This reduced noise and ensured consistency, particularly with images of varying quality. Additionally, GPU memory growth optimization helped the system handle large datasets and training tasks without resource bottlenecks.

The K-Means clustering algorithm was used to analyze similarity scores of clothing images and group them into meaningful clusters for fashion recommendations. The optimal number of clusters was determined using the Elbow Method, and the model efficiently grouped 500 similarity scores, achieving 90% clustering accuracy. The data was split into 70% for training and 30% for testing, with performance evaluated using a classification report and confusion matrix. Images were recommended dynamically based on the input class (e.g., ”Trousers” or ”Sweater”), with visualization tools like PIL and Matplotlib enhancing the recommendation process. The K-Means algorithm demonstrated its value in fashion analysis by improving both clustering accuracy and computational efficiency.

The YOLOv8 model performed excellently in object detection, achieving high accuracy and precision, making it well-suited for identifying clothing items within images. Similarly, DeepLabV3 showed strong performance in segmentation tasks. While its Intersection over Union (IoU) and Dice Coefficient metrics were slightly lower, the model still provided accurate and reliable segmentations. Both models contributed effectively to the system, each excelling in their respective tasks of detection and segmentation.

Despite these advancements, some challenges were observed. Inputs with overlapping or unclear categories occasionally led to recommendation mismatches, such as difficulty distinguishing between formal and semi-formal attire. Additionally, rare or unconventional styles in the dataset contributed to overfitting, limiting the system's ability to adapt to niche fashion preferences. These issues highlight the need to diversify the dataset and improve model interpretability to better handle edge cases.

The system's performance demonstrates the effectiveness of combining vision-based models, similarity search, clustering algorithms, and robust preprocessing strategies. Future work should focus on refining model architectures, expanding the dataset to cover more fashion categories, and incorporating user feedback. These improvements will be crucial for further enhancing the system's accuracy, scalability, and user experience.

VIII. Conclusion

In conclusion, we developed an outfit recommendation system using deep learning techniques. The system combines ResNet50 for feature extraction and a Siamese network with Triplet Loss to provide personalized and accurate outfit suggestions. It features a well-designed backend for seamless model integration, a user-friendly website for interaction, and a scalable database for managing data efficiently.

In the future, we plan to expand the system by adding more clothing categories and improving the recommendation algorithm for faster and more accurate results. We also aim to introduce features like explainability, outfit curation, and filtering options to enhance the user experience. Real-time user feedback and advanced models like Vision Transformers will be explored to further refine the system. These improvements will make the platform a reliable and versatile tool for personalized fashion recommendations.

IX. Project Files & Demo

Dataset: we created a modify version from H&M dataset: kaggle.com/competitions/h-and-m-personalized-fashion-recommendations/data

Demo: here is a live demo link to try Vanilla project: vanillafashion.netlify.app

Frontend: this is the GitHub repository for website frontend: github.com/ReemNawaf/vanilla$_{f}rontend$

Backend: this is the GitHub repository for the project backend containing all the processing functions and the different models: github.com/ReemNawaf/vanilla$_{b}ackend$

X. References

References

[1] Y. Xu, W. Wang, F. Feng, Y. Ma, J. Zhang, and X. He, “Diffusion models for generative outfit recommendation,” *arXiv (Cornell University)*, Feb. 2024. DOI: 10.1145/3626772.3657719. [Online]. Available: https://arxiv.org/abs/2402.17279.

[2] W. Chen, P. Huang, J. Xu, *et al.*, “Pog: Personalized outfit generation for fashion recommendation at alibaba ifashion,” *arXiv (Cornell University)*, Jan. 2019. DOI: 10.48550/arxiv.1905.01866. [Online]. Available: https://arxiv.org/abs/1905.01866.

[3] M. A. Hossen, M. A. Kadir, and N. Alam, “K-means and morphology based feature element extraction technique for clothing patterns and lines,” *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 10, pp. 743–750, 2024. DOI: 10.14569/IJACSA.2024.0151086. [Online]. Available: https://thesai.org/Downloads/Volume15No10/Paper_86-K_Means_and_Morphology_Based_Feature_Element.pdf.

[4] A. Kumar, S. Gupta, R. Mishra, and A. K. Jha, “Fashion recommendation system,” *JETIR (Journal of Emerging Technologies and Innovative Research)*, vol. 10, no. 5, pp. c492–c496, 2023. [Online]. Available: https://www.jetir.org/papers/JETIR2305C92.pdf.

[5] H&M Group, *H&m personalized fashion recommendations*, https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations/data, [Accessed: Nov. 1, 2024], 2024.