AI-Driven Fashion Recommendation System Using Clothing Classification

INFO6550 –Deep Learning | Auburn University Montgomery

Abstract

In this project, we developed an AI-driven system to classify clothing items and recommend visually similar products for online fashion platforms. We used the Fashion MNIST PNG dataset, which includes 70,000 labeled grayscale images across 10 clothing categories.

We trained a custom Convolutional Neural Network (CNN) classifier that achieved 90% accuracy on the test set. The model was designed with image augmentation and early stopping, and it learned to distinguish items like T-shirts, trousers, dresses, and shoes with high confidence. Misclassifications mainly occurred between visually similar upper-body garments, such as shirts and pullovers.

To generate personalized outfit suggestions, we built a content-based recommendation system using ResNet50 pretrained on ImageNet to extract deep visual features. We applied K-Nearest Neighbors (KNN) on these features to identify the top 5 most similar items to any input image.

Our results demonstrate that combining classification with deep feature retrieval can enhance e-commerce product discovery and improve the user experience in digital fashion platforms

Introduction

The fashion industry has seen rapid growth in online retail, but challenges remain in helping customers efficiently navigate thousands of clothing options.

In this project, our team developed a computer vision solution to improve e-commerce product discovery and personalization using AI-driven clothing classification and content-based recommendations.

Our research question was:
"How can deep learning models classify clothing styles and recommend personalized outfits for e-commerce applications?"

We used the publicly available Fashion MNIST PNG dataset consisting of 70,000 grayscale clothing images, grouped into 10 classes (e.g., T-shirt, Sneaker, Dress, Ankle boot).

By combining a Convolutional Neural Network (CNN) for classification and ResNet50-based KNN for similarity retrieval, we created a hybrid system capable of identifying garment types and recommending visually similar products in real time.

This approach can support search optimization, tagging automation, and reduced return rates, while laying a foundation for virtual try-on and other interactive fashion technologies

Literature Review

To ground our work in prior research, we reviewed three key papers that address clothing classification and personalized recommendations in fashion:

- Liu et al. (2016) introduced the DeepFashion dataset and a multitask deep network for fine-grained clothing recognition and attribute prediction. Their work focused on structured retrieval from a massive dataset with rich annotations.
- Han et al. (2017) explored fashion compatibility modeling using Bidirectional LSTMs. Their approach prioritized understanding outfit combinations based on visual and semantic embedding, unlike our single-image, content-based recommendation.
- Guo et al. (2021) proposed a deep learning framework for personalized online clothing recommendations. While their model used user preference data, our project instead relied on visual features alone, extracted from images using ResNet50.

Unlike prior approaches that required either massive multi-attribute datasets or user history, our work offers a lightweight and scalable image-based recommendation pipeline that can run on grayscale input without manual tagging or explicit user feedback.

Study	Dataset	Focus	Key Method
Liu et al.	DeepFashion	Clothing recognition	CNN + attributes
Han et al.	Polyvore	Compatibility modeling	Bi-LSTM
Guo et al.	—	Personalized RecSys	Deep preference network
Ours	Fashion MNIST PNG	Classification + Image similarity	CNN + ResNet50 + KNN

Fig. 0: *Comparison of related studies — our project uniquely combines classification and deep feature-based recommendation using CNN + ResNet50 + KNN on the Fashion MNIST PNG dataset.*

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 6s 18ms/step
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.88      0.85      1000
           1       0.99      0.98      0.99      1000
           2       0.84      0.85      0.84      1000
           3       0.90      0.92      0.91      1000
           4       0.83      0.82      0.83      1000
           5       0.96      0.99      0.97      1000
           6       0.72      0.67      0.69      1000
           7       0.97      0.94      0.95      1000
           8       0.99      0.98      0.98      1000
           9       0.97      0.97      0.97      1000

    accuracy                           0.90     10000
   macro avg       0.90      0.90      0.90     10000
weighted avg       0.90      0.90      0.90     10000
```

Fig. 5: *Classification Report — achieved 90% overall accuracy with strong precision and recall across categories.*

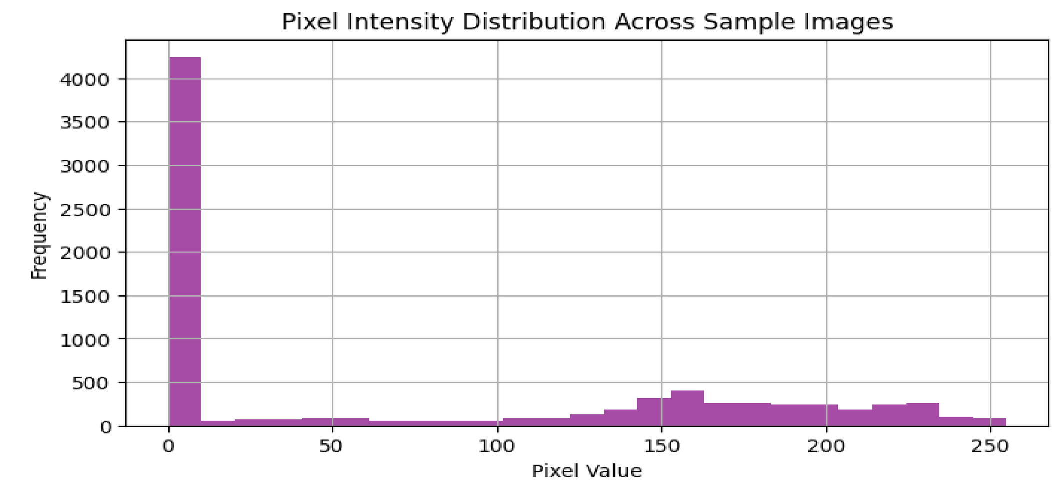

Fig. 1: *Pixel Intensity Distribution Across Sample Images — shows the frequency of pixel brightness levels across grayscale clothing images.*

```
Model: "sequential"
┌───────────────────────────────┬────────────────────────┬───────────────┐
│ Layer (type)                  │ Output Shape           │       Param # │
├───────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d (Conv2D)               │ (None, 26, 26, 32)     │           320 │
│ max_pooling2d (MaxPooling2D)  │ (None, 13, 13, 32)     │             0 │
│ conv2d_1 (Conv2D)             │ (None, 11, 11, 64)     │        18,496 │
│ max_pooling2d_1 (MaxPooling2D)│ (None, 5, 5, 64)       │             0 │
│ flatten (Flatten)             │ (None, 1600)           │             0 │
│ dense (Dense)                 │ (None, 128)            │       204,928 │
│ dropout (Dropout)             │ (None, 128)            │             0 │
│ dense_1 (Dense)               │ (None, 10)             │         1,290 │
└───────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 225,034 (879.04 KB)
 Trainable params: 225,034 (879.04 KB)
 Non-trainable params: 0 (0.00 B)
```

Fig. 2: *CNN Model Architecture Summary — two convolutional layers, followed by dense and dropout layers used for classification.*

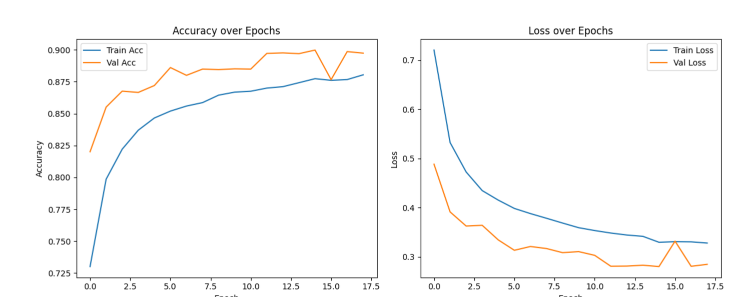

Fig. 3: *Training and Validation Accuracy/Loss — the model steadily improved and reached high validation accuracy with reduced loss.*

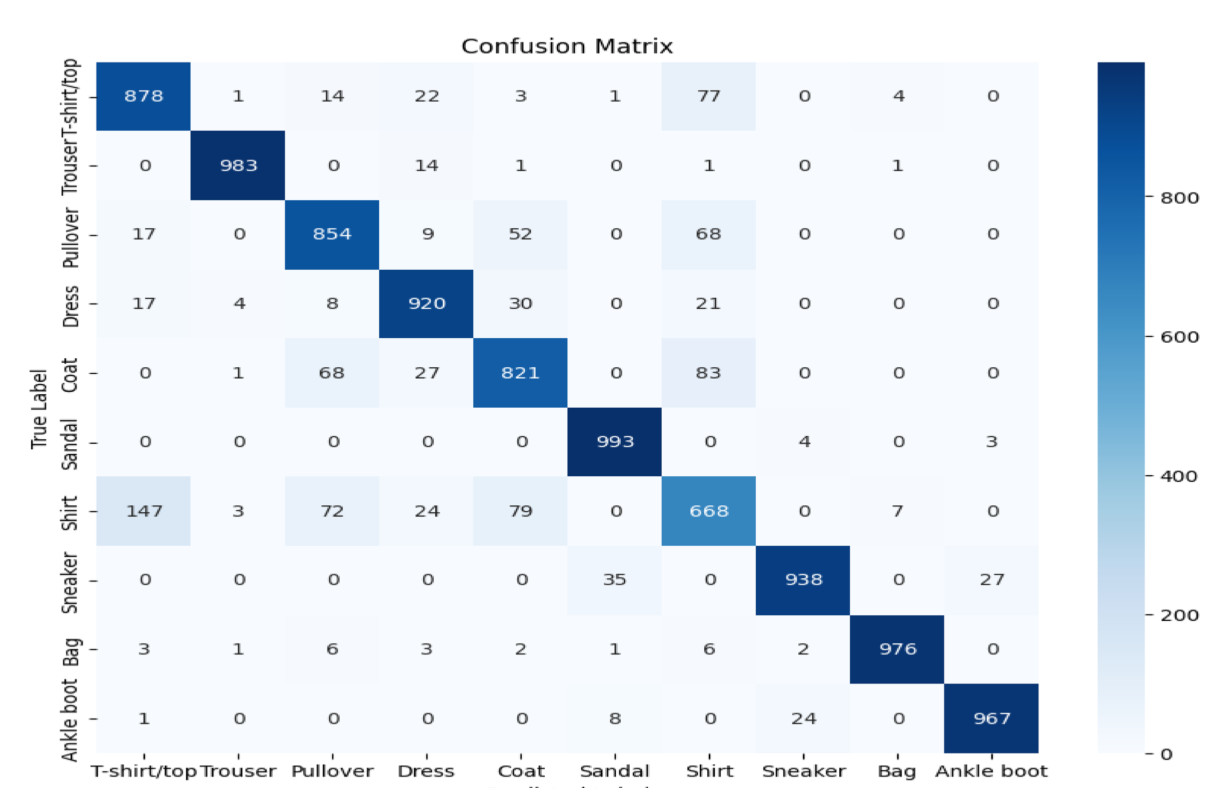

Fig. 4: *Confusion Matrix — high accuracy in most categories, with confusion mainly between similar items like Shirts and Pullovers.*

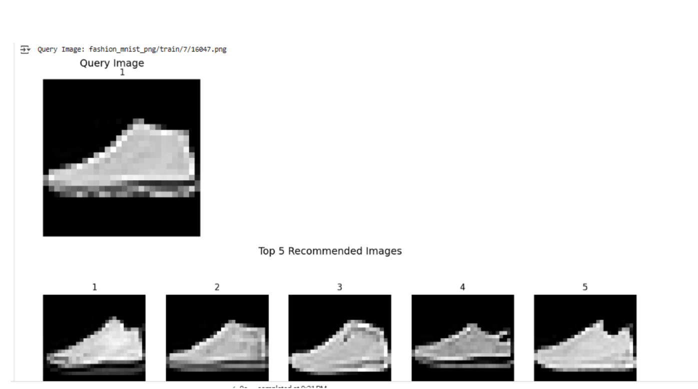

Fig. 6: *Query Image and Top-5 Recommended Matches — retrieved based on visual similarity using deep features + KNN.*

Methodology

Dataset

We used the Fashion MNIST PNG dataset from Kaggle, consisting of 60,000 training images and 10,000 test images. Each image is 28x28 pixels in grayscale, categorized into 10 fashion classes including T-shirt/top, Sneaker, Dress, and Ankle boot.

Preprocessing

All images were rescaled to the [0, 1] pixel range. To improve generalization, we applied data augmentation techniques such as random rotation (±15°), width/height shifting (10%), shearing, and zooming. Labels were derived from directory names and one-hot encoded for classification.

CNN Classifier

We developed a custom Convolutional Neural Network (CNN) using TensorFlow and Keras. The architecture consisted of two convolutional layers with 32 and 64 filters respectively, each followed by max-pooling operations to reduce spatial dimensions. After flattening the output, we added a dense layer with 128 units, followed by a dropout layer (30%) to prevent overfitting. The final output layer used softmax activation to classify the input into one of 10 clothing categories. The model was compiled with categorical crossentropy loss, optimized using Adam, and trained with early stopping to improve generalization.

Feature Extraction with ResNet50

To enable image-based recommendations, we extracted deep features using ResNet50 pretrained on ImageNet, with include_top=False. Grayscale images were replicated into 3 channels and resized to 224x224 RGB. The ResNet outputs were passed through GlobalMaxPooling2D to obtain 2048-dimensional feature vectors.

KNN Recommendation Engine

Using the extracted features, we trained a K-Nearest Neighbors model (k = 6, Euclidean distance) with Scikit-learn. For a given image, the system returned the top 5 most visually similar items, excluding the query itself. Features and file paths were stored in .pkl files for efficient retrieval.

Tools and Environment

We used Google Colab for model development and training. TensorFlow and Keras were used to build the CNN, Scikit-learn for KNN-based recommendations, and Matplotlib, Seaborn, PIL, and OpenCV for visualizations and image processing.

Results & Discussion

We trained our CNN classifier on 60,000 grayscale fashion images using an augmented data pipeline and early stopping to prevent overfitting. The model achieved a final test accuracy of 90%, as reflected in both the classification report and performance plots. The accuracy steadily increased over the epochs, and the loss decreased consistently for both training and validation sets, confirming that the model generalized well on unseen data. Our training plots show convergence around the 15th epoch, with validation accuracy stabilizing just under 90%.

The confusion matrix revealed strong performance on most categories. For instance, the model correctly classified 983 out of 1000 Trousers, 993 Sandals, and 976 Bags — indicating very high confidence on distinct categories like footwear and accessories. However, we also observed confusion between visually similar items. 147 Shirts were misclassified as T-shirts, and Coats were confused with Pullovers or Shirts, likely due to the overlapping shapes and grayscale textures common in upper-body garments. This analysis guided our understanding of where the model could improve, particularly by enhancing representation learning for visually similar categories.

Our classification report confirmed these trends. High-performing classes included Trousers (F1-score: 0.99), Sneakers (0.95), Sandal (0.97), and Bag (0.98). The lowest score occurred in the Shirt class (F1-score: 0.69), which aligns with the misclassification patterns seen earlier. These results suggest the need for more robust features or perhaps higher-resolution inputs to better distinguish ambiguous categories.

To extend our work beyond classification, we implemented a deep feature-based image recommendation system. We used ResNet50 pretrained on ImageNet to extract 2048-dimensional features from the 60,000 training images (converted to 224x224 RGB). Using these embeddings, we trained a K-Nearest Neighbors (KNN) model to recommend similar clothing items based on visual similarity. We validated the recommendation engine by selecting a random query image and retrieving the top 5 most visually similar items. The retrieved products were perceptually coherent, sharing visual structure, outline, or texture with the input image, even though they came from different classes.

These results show that our model not only performs well in multi-class image classification but also enables content-based product discovery — a valuable application in fashion e-commerce.

Conclusion & Future Work

We built a hybrid system that combines CNN-based clothing classification with ResNet50-driven image recommendations. The model performed well on distinct classes like trousers and sandals, but showed confusion between visually similar items such as shirts and pullovers. To improve, future work could explore attention-based models or higher-resolution inputs. The recommendation engine successfully retrieved similar items based on deep visual features. This system can be extended into a personalized recommendation platform by integrating user preferences for dynamic, style-aware product suggestions in fashion e-commerce

References

Fashion MNIST PNG Dataset. Kaggle. https://www.kaggle.com/datasets/andhikawb/fashion-mnist-png

Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). *DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Liu_DeepFashion_Powering_Robust_CVPR_2016_paper.pdf

Han, X., Wu, Z., Jiang, Y.-G., & Davis, L. S. (2017). *Learning Fashion Compatibility with Bidirectional LSTMs*. arXiv preprint. https://arxiv.org/abs/1707.05691

Guo, R., Zhang, Y., & Su, L. (2021). *Deep Learning-Based Clothing Recommendation System for Online Shopping*. arXiv preprint. https://arxiv.org/abs/2101.08301