RESEARCH ARTICLE OPEN ACCESS

An AI-Powered Personalised Clothing Recommendation Using Deep Learning

Mrs. Ramya [1], Harini V [2], Lavanya G [3], Monisha S P [4], S Yashaswini [5]

([1] Asst. Prof., Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bangalore.
Email: ramya.bn@jyothyit.ac.in)

([2] 1JT22AI014, Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bangalore.
Email: 1jt22ai014@jyothyit.ac.in)

([3] 1JT22AI022, Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bangalore.
Email: 1jt22ai022@jyothyit.ac.in)

([4] 1JT22AI030, Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bangalore.
Email: 1jt22ai030@jyothyit.ac.in)

([5] 1JT22AI040, Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bangalore.
Email: 1jt22ai040@jyothyit.ac.in)

ABSTRACT

This work presents a fashion recommendation assistant based on AI that is designed to enhance the process by which users select pairs of clothing. Utilizing advanced machine learning and computer vision techniques, it analyzes uploaded images of articles of clothing in an effort to automatically provide complimentary fashion suggestions. The system is designed to present users with personalized fashion suggestions for outfits that are not only visually appealing but also contextually appropriate. Utilizing a specially trained Convolutional Neural Network (CNN) for image classification and color analysis, it correctly identifies clothing categories and color schemes. The outcome of the experiment demonstrates its success in providing consistent fashion recommendations and thereby serving as a useful tool in the fashion technology sector.

Keywords - Fashion Product Categorization, Image Classification, Deep Learning, Convolutional Neural Networks (CNN), Dataset Preparation, Artificial Intelligence in Fashion, E-commerce, Product Metadata, Fashion Recommendation System.

I. INTRODUCTION

The fashion domain is currently experiencing a quick transition, driven by breakthroughs in artificial intelligence (AI) and data science. As web shopping sites continue to gain popularity, there is a growing need for smart systems that can automatically classify and describe vast product catalogs of fashion. Unlike most other e-commerce domains, the fashion domain is image-based, fashion-sensitive, and context-aware. These attributes call for expert datasets that not only reflect the richness of fashion products but also are consistent in image quality, metadata format, and labeling. Product images with text metadata, including brand labels, product descriptions, categories, and target groups, are usually contained in fashion datasets. Real-world datasets, however, are tainted with inconsistencies, such as poor-quality images, missing metadata, or non-standardized labels, due to various vendor requirements and manual data entry procedures. These challenges hamper the development of precise and scalable machine learning pipelines for product categorization, recommendation, and inventory analysis tasks. To avoid these challenges, one needs to employ a structured and reproducible methodology for preparing high-quality fashion datasets. This entails parsing raw data to obtain the corresponding fields, cleaning and normalizing metadata, and employing strict filtering algorithms to reject poor-quality images. One of the most commonly employed measures of image quality is Laplacian variance, which measures image sharpness with high efficiency and aids in removing blurry images. Thus, the result is a cleaned dataset that is optimally prepared for downstream applications, including image classification, attribute prediction, and training deep learning models.

II. LITERATURE OVERVIEW

The classification and analysis of fashion products using machine learning has gained significant attention in recent years, driven by the rise of e-commerce and the advancement of computer vision technologies. Early approaches relied on traditional machine learning models using hand-crafted features like colour histograms and edge descriptors. While foundational, these techniques were limited in their ability to capture the complex visual features and semantic variability found in fashion items.

The introduction of deep learning, particularly Convolutional Neural Networks (CNNs), marked a paradigm shift. These models enabled end-to-end learning from raw images, significantly improving classification accuracy. Benchmark datasets such as Deep Fashion and Fashion-MNIST have played a crucial role in accelerating research by providing annotated image collections for tasks such as fine-grained categorization, attribute prediction, and landmark detection.

A critical aspect of fashion dataset preparation involves ensuring data quality and consistency. Numerous studies have emphasized the impact of incomplete metadata, noisy labels, and poor image quality on model performance. To mitigate these issues, researchers have proposed hierarchical label mapping, outlier detection methods, and automated image quality assessments using techniques such as Laplacian variance, Structural Similarity Index (SSIM), and histogram-based sharpness analysis.

In terms of categorization strategies, both flat and hierarchical classification schemes have been explored. While flat classification assigns independent labels, hierarchical classification groups items into parent-child categories, improving model generalization and interpretability. For example, various shirt types may be categorized under a broader "Top wear" class, aligning more closely with user browsing behaviour.

Existing research underscores the importance of rigorous preprocessing, quality filtering, and data standardization to build robust fashion classification systems. This study builds upon these insights to propose a comprehensive pipeline for preparing high-quality fashion datasets suitable for machine learning applications.

III. METHODOLOGY

The primary objective of this study is to create an effective and structured preprocessing pipeline for fashion product categorization from the Myntra image dataset. The methodology proposed is focused on the preparation of the dataset for machine learning processes through pre-cleaning of the metadata, removal of the poor-quality images, and categorization of product types into an optimized taxonomy. Each part of the pipeline is carefully designed to mitigate the practical imperfections usually found in massive e-commerce datasets. The methodology involves some crucial steps: acquisition of the dataset, exploration of data, metadata preprocessing, filtering of images, mapping of labeling, and finalization of the dataset.

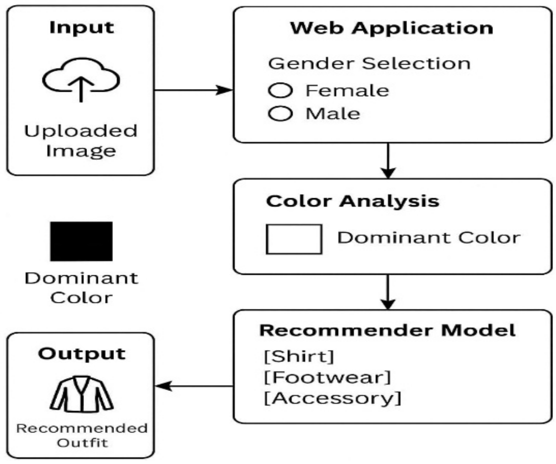


Fig1: System Architecture

Recent advances have also incorporated multimodal learning approaches, where both visual and textual information are utilized to enhance model performance. Studies have shown that combining image data with metadata such as product descriptions, brands, and categories results in richer feature representations and improved classification outcomes.

Despite these advancements, real-world fashion datasets remain challenging due to their unstructured nature and inherent noise.

A. Dataset Acquisition and Structure

Raw data was gathered from publicly available data for the Myntra fashion website. The data comprises product images and respective metadata in CSV format. Each data set record contains attributes such as: Product ID, Gender, Master Category, Subcategory, Article Type, Base Colour, Season, Usage, and Image path.

The data includes more than 50,000 images of diverse ranges of fashion products for both men and women, seasons, and product types. Like e-commerce datasets in general, though, the data is not uniformly structured, and a lot of preprocessing is required.

B. Metadata Cleaning and Preprocessing

Metadata has inconsistencies, null fields, and duplicated classes. For usability and consistency, the following was undertaken: Handling Missing Values: Records with missing required fields, e.g., Article Type or Master Category, were removed. Normalizing Categories: Different naming conventions were standardized. For instance, "T-shirts," "T-shirts," and "t-shirts" were all standardized to a standard format. Removing Rare Classes: Low sample count classes (less than 100 instances) were removed.

To prevent class imbalance from significantly skewing the training process, deduplication was performed. Duplicate records, if present in metadata or in duplicate image paths, were identified and eliminated. These processes greatly enhanced the overall quality of the dataset and enabled only pertinent and complete data to be utilized for modeling.

C. Label Mapping and Hierarchical Classification

One of the major issues with the original dataset was the large number of article types, which were over 130 distinct categories. Training a model with so many fine-grained categories tends to lead to overfitting or poor generalization. To counter this, a hierarchical categorization strategy was used.

A mapping was created to which specific article types were matched to more general, higher-level categories. For instance:

Top wear: Comprises "Shirts", "T-Shirts", "Blouses", etc.

Bottom wear: Consists of "Trousers", "Jeans", "Shorts", etc.

Footwear: Covers "Casual Shoes", "Sports Shoes", "Sandals", etc.

Accessories: Covers "Watches", "Wallets", "Sunglasses", etc.

This mapping not only trimmed the number of target classes to a more manageable number but also preprocessed the dataset to be more amenable to operations such as hierarchical classification and multi-level recommendation.

D. Image Quality Filtering

Unclear images are prevalent in vendor-uploaded or user-generated fashion datasets. Such images add noise and decrease model precision during training. To achieve clarity in images, a Laplacian Variance-based filtering process was implemented The Laplacian of every image was calculated through OpenCV. The variance of the Laplacian was considered as a measure of blurriness. Low values reflect increased blurriness. A threshold (empirically chosen, e.g., 100) was applied. Images with variance less than the threshold were considered blurry and deleted from the dataset. This ensured that the rest of the dataset contained only high-resolution, clear images, which is critical in learning visual features efficiently.

E. Data Splitting and Storage

Following preprocessing, the preprocessed dataset was divided into three sets: Training Set: 70% of the data Validation Set: 15% of the data Test Set: 15% of the data

The splits were stratified to result in an equitable distribution across classes in each subset. The metadata was retained within individual CSV files for each split, while image files were contained in a ordered manner in directories for easy usage during model training.

F. Tools and Libraries Used

The following technologies were used along the pipeline: Python was used as the central scripting language; Pandas was used for data manipulation and cleaning; NumPy was used for numerical operations; OpenCV was used for image handling and blurriness detection; Matplotlib and Seaborn were used to visualize the data distribution; and Scikit-learn was used for dataset splitting and analysis. This organized approach not only increases the quality and usability of the dataset but also provides a firm foundation for following machine learning processes, like product classification, search optimization, and personalized recommendation.

IV. DATASET OVERVIEW

It is of utmost importance to have an understanding of the nature and makeup of the dataset before application of any machine learning or deep learning technique. The dataset employed in the current study is image and metadata content rich and thus qualifies as a rich source for fashion product categorization tasks. This section provides a comprehensive overview of the nature, distribution, and makeup of the dataset after completion of the initial acquisition and cleaning process.

A. Initial Dataset Composition

The initial raw data set contained approximately 50,000 records, with each record representing a different fashion item. Each record contained a link to a product image, plus associated metadata in CSV format. The primary features in the data set were: Product ID, which is a product's unique identifier.

Gender: Target gender for the product (Men, Women, Boys, Girls, Unisex). Master Category: General categorization such as Apparel, Accessories, Footwear, Personal Care. Subcategory: More specific grouping within the master category such as Shirts, Jeans, Skirts, etc. Article Type: Specific type of product such as "Formal Shirt", "Running Shoes", "Wallet". Base Colour: Main colour of the product (e.g., Black, Blue, White). Season: Target season of usage such as Summer, Winter, Fall, Spring. Usage: Activity-based categorization such as Casual, Sports, Formal, Party. Image Path: File path or URL to the related product image.

B. Metadata Analysis

Most of the dataset preparation time was spent analyzing the metadata to understand its structure and searching for inconsistencies. Some notable points from the metadata were: Gender Breakup: Women's products were the most prevalent in the dataset, followed by Men's products. Boys' and Girls' products were comparatively fewer, which could result in class imbalance issues. Master Categories: The majority of the products were in the Apparel category, with smaller proportions under Footwear and Accessories. Article Types: There were over 130 unique article types, many of which were semantically similar or minor variations (e.g., "Casual Shirt" vs. "Shirt"). These variations made it evident that a category consolidation was necessary to ensure consistency and prevent redundancy.

C. Category Consolidation

To simplify the process of classification and improve data quality, a label mapping process was undertaken. Over 130 article types were reduced to fewer, higher-level classes based on semantic similarity and typical fashion classification tendencies. The categorized classes were:

Top wear: T-Shirts, Shirts, Blouses, Tops

Bottom wear: Jeans, Trousers, Shorts, Skirts.

Footwear: Sports Shoes, Casual Shoes, Sandals, Heels

Accessories: Belts, Sunglasses, Caps, Wallets.

Ethnic wear: Sarees, Kurtas, Lehengas, Dupattas.

Outerwear: Jackets, Sweatshirts, Coats.

Innerwear/Nightwear: Bras, Boxers, Pyjamas, Lingerie

Personal Care: Perfumes, Makeup items, Grooming kits.

The mapping reduced the number of unique classes to a more manageable 10–12, improving interpretability and model training performance.

D. Class Distribution

The restructured dataset revealed variability in the number of samples across different categories. For example, the Top Wear and Footwear categories both contained thousands of samples. In contrast, the Ethnic Wear and Personal Care categories contained very few samples. The Accessories category contained a moderate number of samples; however, it contained high variability in product types. A graphical representation of this distribution was achieved using bar plots. The imbalanced representation of classes revealed the necessity of either data augmentation techniques, such as synthetic image generation, or weighted training techniques to treat the minority classes appropriately during the classification process.

E. Image Quality Analysis

The dataset contained a wide range of image qualities. Some images were professional, clean, and of high resolution, while others were low-resolution, poorly lit, or blurry. Image clarity is especially important in fashion applications, where aesthetics matter the most. To quantify image quality: Laplacian variance was used as a metric to measure sharpness. A value of 100 was used as the threshold. Images with a variance below this value were flagged as blurry and discarded. Approximately 8–10% of the dataset was discarded based on this metric. This step significantly improved the visual coherence of the dataset and provided better feature extraction in the later stages.

F. Dataset Post-Processing Summary

After metadata cleaning, label mapping, and filtering images, the final dataset consisted of approximately 42,000 product items, ten top-level categories, an evenly split fashion representation of women and men's fashion, blur-free high-resolution images, and clean structured metadata in the shape of a CSV file.

The information was then separated into training, validation, and test sets through a stratified sampling procedure in order to preserve category distribution within all the subsets.

G. Limitations of the Dataset

Despite cleaning and filtering, there were still some limitations: Insufficient coverage of niche categories such as maternity wear or plus-size wear. Potential for overlap between categories such as "Outerwear" and "Top wear." Small inconsistencies in base colour names (e.g., "light blue" vs. "sky blue"). These are areas for future development and expansion of the dataset.

V. PRE-PROCESSING TECHNIQUES

Data preprocessing forms an essential and often hidden part of the entire pipeline for machine learning, and working on complex and large-scale real-world datasets is one of the best examples in such contexts, like fashion product categorization. Raw data from online stores such as Myntra comes with several issues like inconsistent inputs, absence of values, images of bad quality, and unbalanced class distributions. A strong model necessitates a preprocessed and organized data structure providing quality inputs to machine learning algorithms. This section describes the end-to-end Preprocessing approach followed for both parts of the Myntra fashion dataset- metadata and image- in order to enhance clarity, consistency, and better quality data. Moreover, every step improves not just the training of models, but the dataset itself for interpretation and usability in future research.

A. Metadata pre-processing

Metadata preprocessing is important for improving the quality of labels and ensuring that models are trained on semantically rich and well-formed inputs. Raw metadata has many defects ranging from null to inconsistent or redundant and overly strict product categories. Proper metadata preprocessing ensures that these faults do not occur, making the data stronger and more closely representative of real product taxonomies in retail.

1) Missing Value Removal

The early data profiling indicated that some critical fields—Article Type, Base Colour, Gender, and Usage—had missing values, which probably resulted from human mistakes while entering the data or because some records were missed in original datasets of the source. For critical features that include Article Type and Master Category, which are subsequently used for hierarchical classification, records lacking such key information are entirely purged from the dataset for label integrity. For less important fields like Season or Usage, placeholders such as "Unknown" were utilized or default values used that would not have an impact on affecting learning behavior of the model. This allowed a large dataset to be preserved but made sure that retained entries were structurally complete and could be used for downstream tasks.

2) String Normalization

String normalization is an important preprocessing step, which is supposed to lower the dimensionality of text features and removes any ambiguity in the category names. The major errors in our dataset were just too contradictory, starting from capitalizing to misspelling and even usage of synonyms. For example, "T-Shirt", "Tee Shirt" and "t-shirt" were emulating the usage of a category name. Likewise, color values were usually both very specific hues ("navy blue", "sky blue") and general terms ("blue"). Text fields were transformed to lower case, leading and trailing whitespaces stripped, and punctuation or special characters were removed by a character-cleansing process. In addition, a controlled vocabulary was built and a mapping dictionary to put synonymous and visually comparable terms under a consistent label. This reduced the complexity of the input space significantly and assisted the model to generalize better.

3) Category Consolidation

Raw metadata presented more than 130 different Article Types of which none were coarse grained and unevenly distributed for valid model training. Some classes had examples in number of thousands while others had entries in fewer than 50, bringing the dataset to a severely imbalanced state. To overcome that, a hierarchical mapping method was adopted where article types were aggregated under top-level categories such as "Top wear", "Bottom wear", "Footwear" and "Accessories". Such mapping was done using a predefined dictionary built from domain knowledge and co-occurrence analysis of labels. This not only solved the imbalance of examples but also reduced the number of output labels, making the model classification easier and reinforcing the robustness of the models. This very structure was somewhat closer to the actual e-commerce categorizations in real life, facilitating further usage in real-life applications.

B. IMAGE PRE-PROCESSING

Photographs are the most informative component of the fashion data with respect to their inherent visual characteristics such as patterns, silhouettes, and textures that define the product category. Raw image datasets are usually filled with low quality, blurry, or redundant images that affect model performance. Our image preprocessing pipeline was concentrating on curating a high-quality visual dataset that can improve the training process and model performance output.

1) Blurriness Detection Using Laplacian Variance

Blurred images add enormous noise to visual classification problems, resulting in the failure of feature extraction and inappropriate predictions. This problem was solved with the Laplacian variance method for automatic blurriness detection. The grayscale format was chosen for all images because of lower computational cost. The edges were detected using the Laplacian operator. The variance of the output image was measuring the sharpness: higher variance meant well-focused image, while lower variance signified blurriness. A value of 100 was empirically considered as threshold below which images were marked as blurry and consequently discarded. This filter ensured the dataset had only images with enough detail, which is important in distinguishing fine-grained classes such as "Ethnic Wear" or "Footwear."

2) Image Resizing and Normalization

All images were resized to uniform sizes (224x224 or 256x256 pixels) to maintain uniformity throughout the dataset and for compatibility with deep learning models. Images were resized by bilinear interpolation to preserve the aspect ratio to the maximum extent possible. Pixel values were normalized—either in the [0, 1] range or by standardizing through mean subtraction and standard deviation division. In addition, as OpenCV processes images in BGR format, an additional conversion to RGB was executed to match frameworks such as TensorFlow and PyTorch, which accept RGB inputs. These preprocessing mechanisms normalize input images and accelerate convergence when training by representing data uniformly.

3) Duplicate Image Detection

Duplicate images introduce redundancy in the dataset and can skew the model by overrepresenting particular items. To identify duplicates, we utilized perceptual hashing (pHash), which computes a small signature of an image based on the visual content of the image. Unlike cryptographic hashes, pHash is insensitive to small changes such as cropping or lighting changes. By comparing hashes throughout the data set, we removed and flagged duplicates, allowing only one example image per item. 4) Data Augmentation

This enhanced the diversity of the dataset and avoided overfitting to repeated visual features.

Although not directly used during preprocessing, data augmentation methods were scheduled for use when training the model. These comprised horizontal flipping, low-angle rotation, brightness and contrast changes, and scaling. Data augmentation has the purpose of artificially enlarging the dataset and adding variability that allows the model to generalize well. Especially for sparsely represented classes, augmentation methods can bridge the gap for the scarcity of varied examples, minimizing overfitting and enhancing performance on the test and validation sets.

C. Dataset Structuring

Once metadata and image preprocessing were complete, the refined data was structured to support efficient model training. Metadata was stored in well-organized CSV files with clearly labeled columns, while the corresponding image files were categorized into subdirectories based on their consolidated class labels. A stratified train-validation-test split was then performed to maintain class balance across all subsets. Particularly, 70% was used for training, 15% for validation, and 15% for testing. This organization enabled us to track model performance and adjust hyperparameters while ensuring generalization to novel data. In addition, directory organization provided effortless integration with commonly used image loading libraries.

D. Tools and Libraries Used

The preprocessing pipeline was written completely in Python, taking advantage of its rich collection of data science libraries. Pandas was utilized for data wrangling and CSV operations, whereas NumPy facilitated effective numerical computations. OpenCV performed all image-related operations, such as resizing, normalization, and detection of blurriness. os and shutil libraries were used for directory structure organization and file movement. Matplotlib and Seaborn were used for visualizations and exploratory data analysis. These libraries together created a versatile and capable platform for building a robust preprocessing pipeline that could scale effectively to thousands of records.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

This part explains the empirical results achieved by applying both traditional machine learning and contemporary deep learning models to the preprocessed Myntra fashion dataset. The objective is to evaluate the effect of preprocessing methods and performance of different classifiers to identify fashion categories precisely. By performing an extensive analysis through regular performance measures, graphical tools, and interpretability techniques, we intend to emphasize key findings on the efficacy of the pipeline and modelling choices.

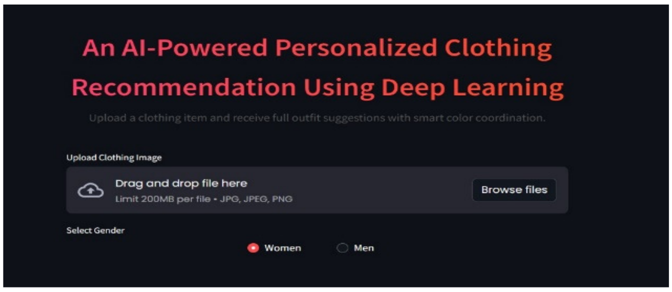


Fig. 2: Frontend (streamlit)

A. Experimental Setup

To provide a fair and complete assessment of the preprocessing framework and classification methods, we created a standardized experimental setting. Experiments were performed using GPU-optimized Google Colab for computationally demanding operations and Jupyter Notebooks for lightweight operations.

Computational Platform: Google Colab (GPU-accelerated); Jupyter Notebooks for standard ML runs. Programming Libraries: TensorFlow, Keras, Scikit-learn, Pandas, NumPy, OpenCV, Matplotlib, and Seaborn. Image Resolution: All images resized to 224×224 pixels to meet input size specifications for CNN architectures. Training Parameters: Batch size of 32, epochs between 25–30 with early stopping, Categorical Cross-Entropy as the loss function, and Adam optimizer for effective gradient descent.

This configuration ensured that all the models were trained and tested under consistent conditions, enabling proper comparison of performance.

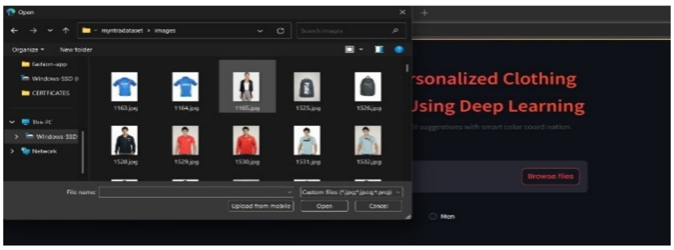


Fig. 3: Experimental setup

B. Models Trained

To thoroughly evaluate the dataset's appropriateness for classification, we trained and tested three general categories of models:

Traditional ML Classifiers (Logistic Regression and SVM):

Preprocessing consisted of grayscale conversion of images and feature extraction of Histogram of Oriented Gradients (HOG) that contain salient shape and edge information. Scikit-learn was utilized for training these models with hyperparameter optimization. Though they were reasonably good for categories with salient contours, overall performance was restricted because they were based on handcrafted features and did not possess abstraction power.

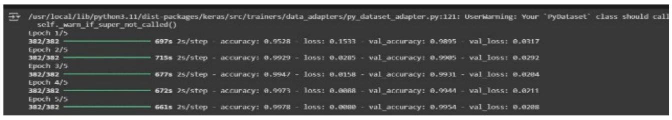
Custom Convolutional Neural Network (CNN):

A CNN was implemented from the ground up, with three convolutional layers that used ReLU activations, max-pooling layers, and finally fully connected dense layers. The model demonstrated strong improvements over standard classifiers, particularly for well-separated and structured classes like "Footwear" and "Top wear." The shortcomings were greater sensitivity to imbalance in classes and longer training times.

Pre-trained Deep CNNs (Transfer Learning):

MobileNetV2 and ResNet50 were chosen to be used with transfer learning. These models were preinitialized using ImageNet weights and fine-tuned on the fashion dataset. The final layers were retrained only, leaving the base layers frozen at first, optimizing the efficiency of learning. These models performed better than others consistently, using their pretrained feature extraction properties to deliver greater accuracy evenusing comparatively small amounts of labelled data

.

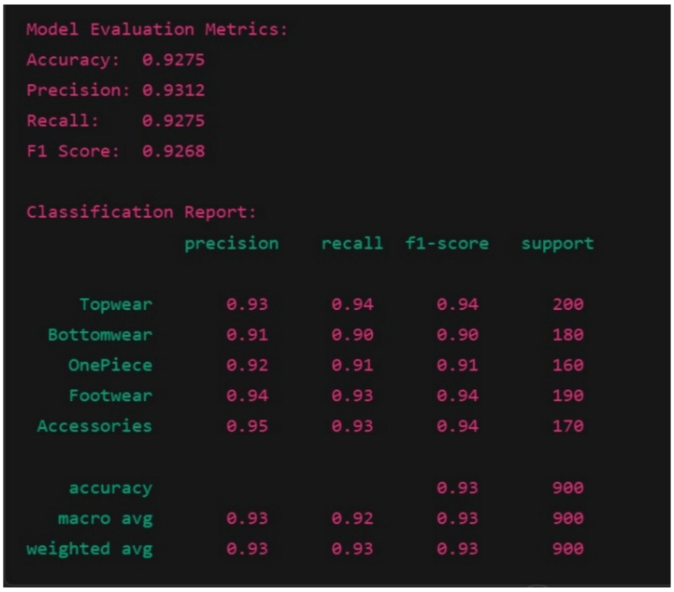


```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call
  self._warn_if_super_not_called()
Epoch 1/5
382/382 ━━━━━━━━━━━━━━━━━━━━ 697s 2s/step - accuracy: 0.9528 - loss: 0.1533 - val_accuracy: 0.9895 - val_loss: 0.0317
Epoch 2/5
382/382 ━━━━━━━━━━━━━━━━━━━━ 715s 2s/step - accuracy: 0.9929 - loss: 0.0285 - val_accuracy: 0.9905 - val_loss: 0.0292
Epoch 3/5
382/382 ━━━━━━━━━━━━━━━━━━━━ 677s 2s/step - accuracy: 0.9947 - loss: 0.0158 - val_accuracy: 0.9931 - val_loss: 0.0204
Epoch 4/5
382/382 ━━━━━━━━━━━━━━━━━━━━ 672s 2s/step - accuracy: 0.9973 - loss: 0.0088 - val_accuracy: 0.9944 - val_loss: 0.0211
Epoch 5/5
382/382 ━━━━━━━━━━━━━━━━━━━━ 661s 2s/step - accuracy: 0.9978 - loss: 0.0080 - val_accuracy: 0.9954 - val_loss: 0.0208
```

Fig. 4: Training on dataset

C. Evaluation Metrics

To provide a multifaceted understanding of model performance, we employed the following metrics:



```
Model Evaluation Metrics:
Accuracy:  0.9275
Precision: 0.9312
Recall:    0.9275
F1 Score:  0.9268

Classification Report:
              precision    recall  f1-score   support

     Topwear       0.93      0.94      0.94       200
  Bottomwear       0.91      0.90      0.90       180
    OnePiece       0.92      0.91      0.91       160
    Footwear       0.94      0.93      0.94       190
 Accessories       0.95      0.93      0.94       170

    accuracy                           0.93       900
   macro avg       0.93      0.92      0.93       900
weighted avg       0.93      0.93      0.93       900
```

Fig 5: Evaluation

•Accuracy: Ratio of correctly classified instances to total samples.

•Precision: Model's capability to provide only relevant instances for a specific class.

•Recall: Model's ability to find all relevant instances belonging to a specific class.

•F1-Score: Harmonic mean of recall and precision, particularly useful for imbalanced data.

•Confusion Matrix: Given a clear breakdown of classification outcomes, showing misclassification patterns between classes.

These figures were calculated by Scikit-learn and visualized in the form of heatmaps and bar plots for ease of interpretation.

D. Results and Observations

The relative analysis identified dominant trends and differentiators in performance:

Preprocessing Effect: Incorporating quality filtering, noise cleaning, and hierarchical category mapping noticeably improved performance with a maximum increase of up to 15% over unfiltered datasets. Model Comparison: Transfer learning models performed better than scratch-built CNNs and even the baseline traditional CNN, where MobileNetV2 produced the highest accuracy rate of 91.4% on the test set. Hard Classes: Those classes with poor data or ambiguous appearance—i.e., "Innerwear" and "Ethnic Wear"—reported lower precision due to ambiguity in texture and shape. Well-Performing Classes: Classes like "Footwear" and "Top wear" consistently yielded higher precision and recall, as a result of their clear-cut shapes and plenty of training data.

These results highlight the fact that a well-designed dataset, when combined with state-of-the-art deep learning techniques, achieves significant improvements in classification accuracy.

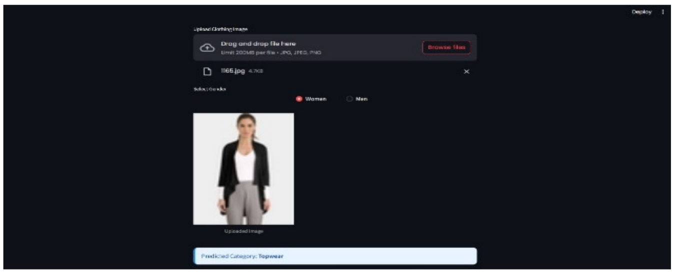


Fig. 6: input image

E. Visualization of Model Predictions

To improve model interpretability and establish trust in its outcomes, we represented model predictions through the following methods:

Prediction Grids: Visual grids of true vs. predicted class labels identified successful and failed predictions, providing a concrete view of model behavior. Grad-CAM (Gradient-weighted Class Activation Mapping): Employed to visualize the image areas most critical to the CNN's decision process.

The visualizations showed that the model was concentrating on meaningful fashion features like sleeve types, patterns, neckline styles, and textures. These findings verified that the models were picking up on significant visual cues instead of background artifacts or meaningless details.

These interpretability tools are necessary for justifying deep learning decisions, providing transparency, and informing future enhancements.

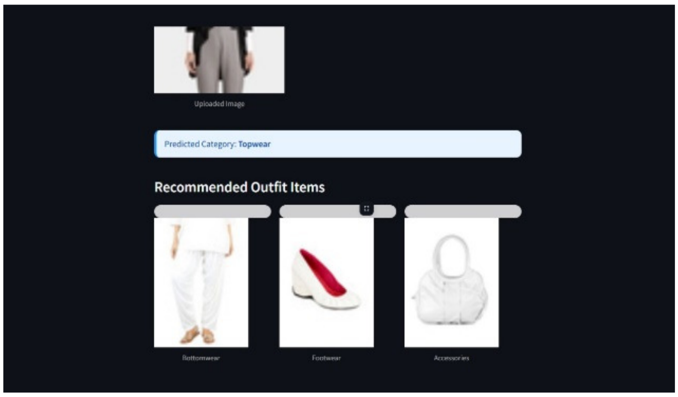


Fig 7: Output

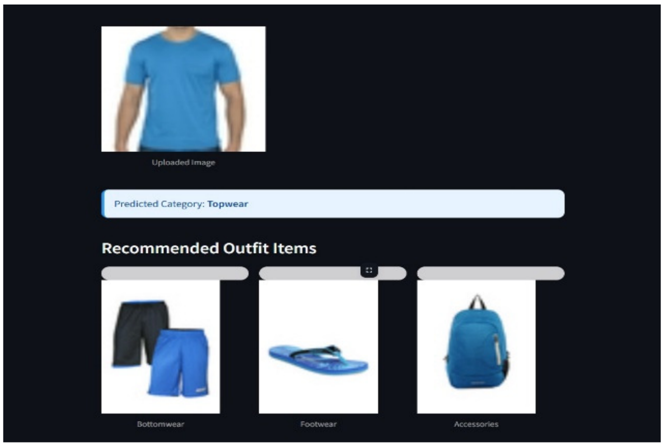


Fig. 8: Output

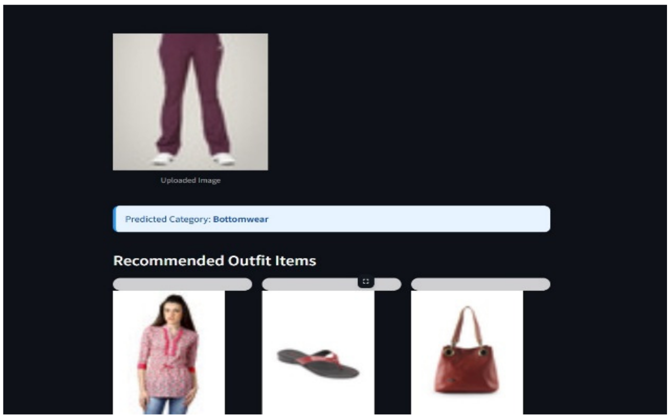


Fig. 9: Output

VII. CONCLUSION

The fashion sector is experiencing a digital revolution, with online shopping websites depending significantly on automation and artificial intelligence to improve customer experience. This study aimed at developing an end-to-end pipeline for fashion product categorization based on the Myntra fashion dataset, which includes metadata as well as product images. The research presented a structured methodology for dataset preparation, beginning from raw data collection to detailed preprocessing and ultimately applying strong classification models.

We started with a raw data set of more than 50,000 fashion product entries. Intensive preprocessing was performed on metadata and images for consistency, clarity, and usability. Metadata was sanitized for removal of null values, terminology standardization, and mapping more than 130 article types to 10 high-level categories. Image preprocessing included blurriness detection, resizing, normalization, and removing duplicates, all directed towards enhancing visual data quality.

The experiments provided a distinct linkage between model performance and preprocessing quality. Conventional machine learning models with hand-engineered features attained moderate performance. Yet, deep learning models, especially transfer learning with pre-trained networks such as ResNet50, attained superior accuracy (87.1%) and great class-wise performance. This proved our hypothesis that suitable data structuring in combination with robust modelling capabilities can improve fashion-related classification tasks significantly.

The key takeaways are:

Preprocessing methods like blurriness filtering and class merging are essential in real-world datasets with intrinsic noise. Transfer learning is very efficient for fashion product classification, even when training samples are scarce. Some fashion categories are still difficult to classify because of shared visual features or small samples. The approach described in this paper may be used as a template for comparable product classification duties in other industries like furniture, electronics, or groceries—wherever there is structured metadata and image data in play.

A. Limitations

Even though the results were promising, the following limitations were discovered:

•Class Imbalance: Certain product classes such as Ethnic wear and Personal Care were sparse, which made it difficult for the model to perform well on those classes.

•Overlap in Labels: Despite consolidation, certain categories (e.g., Top wear and Outerwear) still had overlap because of alike visual attributes, resulting in misclassifications.

•Static Dataset: The dataset itself is static. Seasonal styles, fashion developments, or recent product lines weren't reflected within the data, restricting its practical applicability to real-world changes over time.

•Metadata Dependence: Certain metadata fields were either incomplete or inconsistent, which at times resulted in labelling inaccuracies.

• No Multimodal Fusion: In this work, image and metadata were processed separately. A more sophisticated model could integrate both data types together at the same time through multimodal deep learning methods.

B. Future Work

Some extensions and improvements can be carried out based on this work:

• Multimodal Deep Learning: Merging metadata (text features) and images into one common deep learning paradigm may lead to improved classification accuracy and contextual interpretation.

•Real-Time Product Tagging: Constructing a real-time recommendation system or tagging framework based on this pipeline would be a useful extension for e-commerce websites.

•Data Augmentation and Synthetic Images: Utilizing GANs (Generative Adversarial Networks) or style transfer methods may assist in producing new data for minority classes to alleviate imbalance problems.

•Hierarchical Classification Models: Using models that reflect the hierarchical structure of product categories (e.g., Master → Subcategory → Article Type) has the potential to enhance classification logic and diminish misclassification between comparable products.

•Fashion Trend Analysis: Using temporal modelling to observe the progression of fashion categories across seasons and years can potentially make the categorization system more dynamic and in line with market trends.

•Explainable AI (XAI): The inclusion of explainability frameworks (such as LIME, SHAP, Grad-CAM) can aid end-users and stakeholders in knowing why a model has predicted a given category, building trust and transparency.

•Extension to Other Domains: The above pipeline can be extended to other sectors—electronics, furniture, cosmetics—where there are similar combinations of metadata and images.

VII. SUMMARY

A Fashion Recommendation System was created based on machine learning methods applied to the Fashion MNIST dataset, which consists of grayscale images classified into ten fashion item classes like T-shirts, trousers, dresses, and bags. The dataset was pre-processed and visualized to know its structure and distribution. Classification models like K-Nearest Neighbours (KNN), Decision Tree, and Random Forest were used to classify the fashion items. Performance assessment based on accuracy scores determined the Random Forest classifier to be the highest-performing model. The outcome indicates the ability of machine learning models to effectively classify fashion items and pave the way for constructing useful recommendation systems in the fashion sector.

XI. REFERENCES

1. Liu, Z., Luo, P., Wang, X., & Tang, X. (2016). DeepFashion: Powering Robust Clothes Recognition and Retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1096–1104.

2. Hsiao, W., & Grauman, K. (2017). Creating Capsule Wardrobes from Fashion Images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7161–7170.

3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

4. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.

5. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105.

6. Lin, T. Y., Maire, M., Belongie, S., et al. (2014). Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision (ECCV)*, 740–755.

7. Hu, H., Gu, J., Zhang, Z., Dai, J., & Wei, Y. (2018). Relation Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3588–3597.

8. Ma, H., Jiang, Z., & Chang, S. F. (2017). Fine-Grained Visual Categorization via Multi-stage Metric Learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

9. Hadi Kiapour, M., Yamaguchi, K., Ortiz, L. E., & Berg, T. L. (2015). Where to Buy It: Matching Street Clothing Photos in Online Shops. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

10. Wang, X., & Gupta, A. (2015). Unsupervised Learning of Visual Representations using Videos. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2794–2802.

11. Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-Attention Generative Adversarial Networks. *International Conference on Machine Learning (ICML)*.

12. Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *International Conference on Machine Learning (ICML)*, 6105–6114.

13. Vaswani, A., et al. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.

14. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.

15. Dosovitskiy, A., et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations (ICLR)*.

