Northwestern

Fashion Product Images Analysis

Deep Learning Final Project - Winter 2025

Team 02: Haowen Geng, Runxuan Li,
Ruyi Lu, Wenfei Wu, Wenyang Miao

Executive Summary

A relevant fashion-related problem with AI-generated fashion classification visual.

Outcome: A data-driven approach to **classify clothing images into categories** with high accuracy, while applying **style transfer to clothing items based on sample style**.

Problems

Image Quality
Images compressed making its poor quality

Training discrepancies
need careful tuning for optimal convergence

Data Imbalance
Some Classes have more samples than other

Model training time
Slow due to iterative optimisation

Strategies

Pipeline of Deep Learning Analysis:

Data Preprocess
Image Loading & normalization

Feature Engineering
Resizing & extract content style separately

Deep learning method
Segmentation & Feature Encoder

Post-processing
reintegrates results to original image

Analysis

Classification Models: MobileNet, ResNet50, EfficientB0, self-trained model
Style Transfer Models: TensorFlow Hub Style Transfer, U-Net with MobileNetV2, & self-trained NST with VGG-19

Dataset Overview & Data Cleaning Steps

Discuss project scope and preliminary steps taken

The fashion product dataset consists of approximately **44,000+ fashion product images with labels,** including its category, color, seasonality, and intended usage.

Three key objectives:

1. **Classification:** Classify fashion product images into categories such as apparel, accessories…
2. **Style Transfer:** Generate creative apparel designs using image transfer
3. **Personalization:** Recommendation of clothing based on current outfit (optional)

Steps Performed: Data Preprocessing

1. **Image Loading**

- The `load_image2()` function loads images using `PIL` and converts into RGB format.
- The images are transformed into tensors and normalized before further processing.

2. Segmentation Mask Extraction

- A segmentation model (**DeepLabV2**) is used to extract the T-shirt mask from an input
- The mask is generated using the model's **argmax() output**, identifying highest probability class for each pixel.
- A binary mask is created, where **T-shirt pixels are set to 255** and non-T-shirt regions are set to 0. This enables precise **targeted style application**.

Exploratory Data Analysis

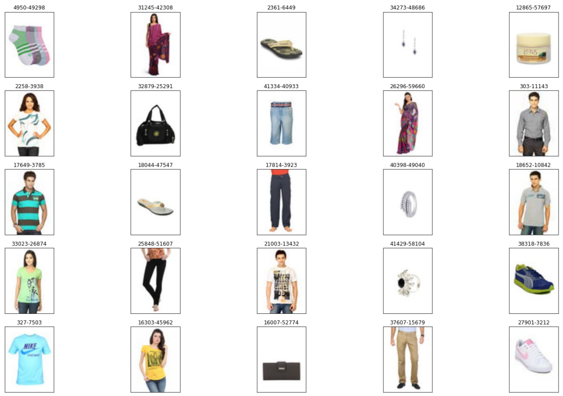
Identify missing value with data overview

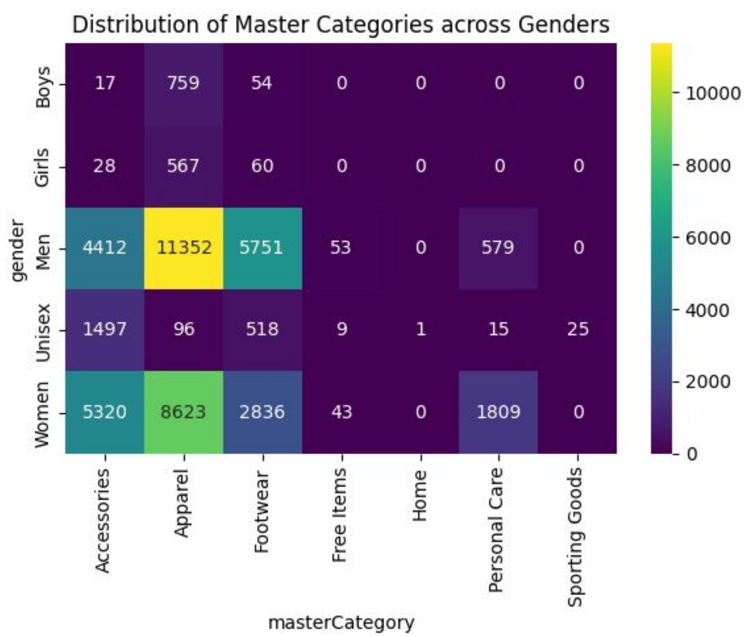
The dataset has **9 features**, with few missing values in some columns.

Features with the **most missing values**:

- **Usage (317 missing)**
- **Base-colour (15 missing)**
- **Season (21 missing)**

```
Missing Values:
id                    0
gender                0
masterCategory        0
subCategory           0
articleType           0
baseColour           15
season               21
year                  1
usage               317
productDisplayName    7
dtype: int64
```





Preliminary Insights

1. **Master Category**: Dominated by **Apparel**, followed by **Footwear** and **Accessories**. Other categories have minimal representation.
2. **Free Items, Home, Personal Care, and Sporting Goods** appear to have minimal representation, with very few occurrences across the groups.
3. The distribution suggests that **certain categories dominate** overall sales or records, while others are much less common.

Image Classification

Result of EfficientNetB0

1. Data preprocessing

- Encode labels for selected categories (Apparel, Footwear, Accessories)
- Resize Image to 128x128 for model compatibility

2. Training Configuration

- **Optimizer:** Adam
- **Loss Function:** Categorical Cross Entropy
- **Metric:** Accuracy

3. Model Architecture

Base Model:

- EfficientNetB0

Additional Layers

- GlobalAveragePooling2D to reduce dimensions.
- Dense (128 units, ReLU activation) for feature extraction
- Dropout (0.5) to prevent overfitting
- Dense (final layer, softmax activation) for classification

4. Testing Results

Final accuracy: **0.51**

Model Architecture: EfficientNetB0-based Image Classifier

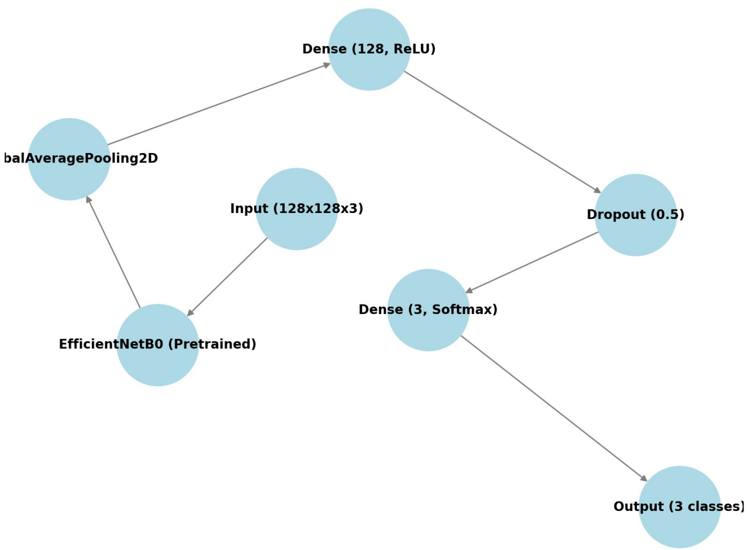


Image Classification

Discuss why self-trained model, MobileNetV2, and ResNet50 achieving much higher accuracy

1. **Employed the same data preprocessing steps and training configurations**
2. **Pre-trained Model Architecture: Base model + Additional Layers**
   - GlobalAveragePooling2D to reduce dimensions
   - Dense (1024 units, ReLU activation) for feature extraction

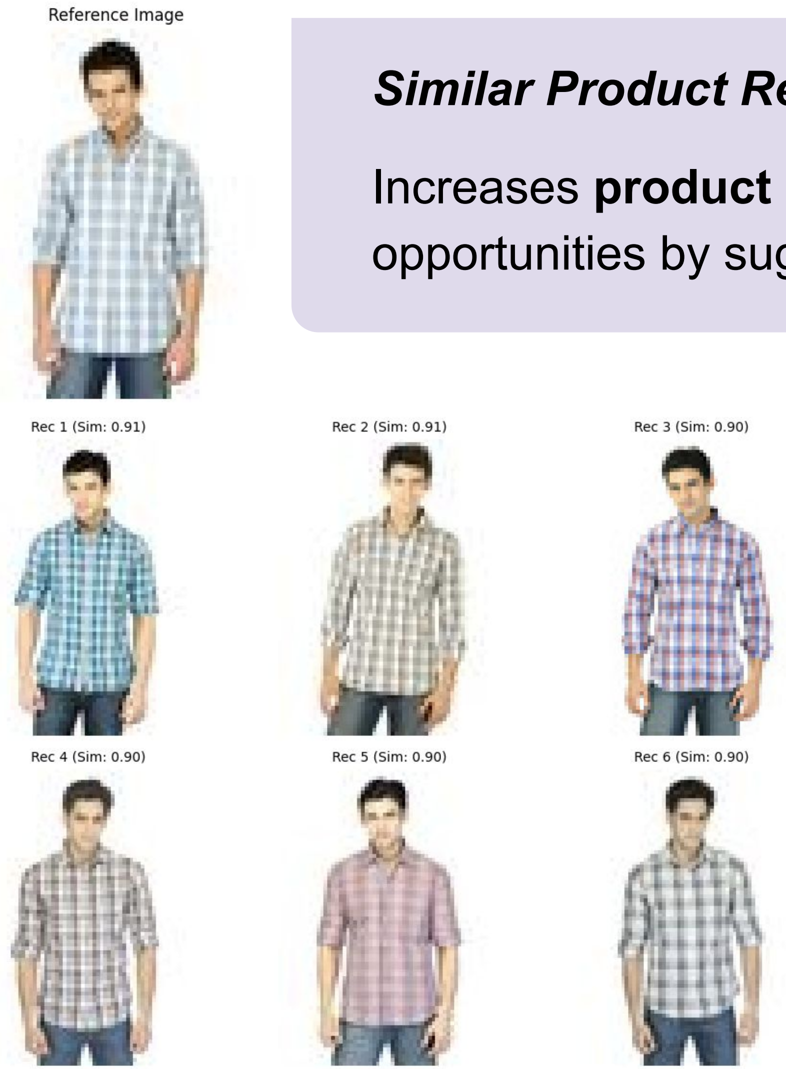
	Test Accuracy
MobileNetV2	**99.25%**
Self-trained *model architecture in appendix 2	**99.00%**
ResNet50	**95.92%**
EfficientNetB0	**51.04%**

Possible Reasons for Discrepancy in Accuracy:

- **Model Depth** → Dense layer with 1024 vs 128 nodes allows for **richer feature extraction** to capture complex relationships in the data.
- **Dropout layers** → While dropout helps prevent overfitting, it also **reduces the model capacity.**
- **Balanced Scaling Strategy** → EfficientNet scales depth, width, and resolution uniformly, which might not be the optimal configuration for this dataset.

Image Classification

ResNet feature extraction to similar product recommender



Similar Product Recommender

Increases **product discovery** and **cross-selling** opportunities by suggesting visually similar alternatives.

Embed

Extract Feature Embeddings Using ResNet

Execute

Compute Similarity Score Using Cosine Similarity

Style Transfer – Goal and Methodology

Overview of Style Transfer Study

Goal:

- To integrate patterns from famous painting to fashion designs while preserving structural details.

sample style for style transfer model

Methodology

- Style transfer application with/without segmentations
- Pattern extraction of painting from Vincent Van Gogh as sample style
- Fashion integration of pattern extracted and more basic design clothing

sample images for style transfer

Style Transfer – Model Overview

Architecture of all models applied for model comparison

01 MobileNetV2-based feature extraction + Deep Learning-based Style Transfer

- **Segmentation:** None
- **Feature Extraction**: MobileNetV2
- **Style Transfer**: Magenta Arbitrary Stylization Model (TensorFlow Hub)
- **Description**: Uses **Magenta Arbitrary Stylization Model** to apply a pre-trained style transformation to the entire image

02 Hybrid Stylization Model: DeepLabV3-based segmentation

- **Segmentation:** DeepLabV3+ (ResNet101)
- **Feature Extraction:** MobileNetV2
- **Style Transfer: Magenta Arbitrary Stylization Model (TensorFlow Hub)**
- **Description:** Uses **DeepLabV3+** to provide a mask for the clothes region, **MobileNetV2** used for feature extraction, and **Magenta Arbitrary Stylization Model** to apply style transformation

Style Transfer – Model Overview Continued

Architecture of all models applied for model comparison

03

Hybrid Stylization Model: UNet-based segmentation

- **Segmentation**: UNet with MobileNetV2 as encoder
- **Feature Extraction**: MobileNetV2
- **Style Transfer**: Magenta Arbitrary Stylization Model (TensorFlow Hub)
- **Description**: Uses **UNet** for **fine-grained pixel-wise segmentation** of the clothes, with **MobileNetV2** as the encoder backbone in UNet

04

Self-Trained Neural Style Transfer Model using VGG-19

- **Feature Extraction**: VGG-19 (Pretrained on ImageNet)
- **Style Transfer**: Custom-trained using **content loss and style loss** (with Gram matrix method)
- **Description**: Uses **VGG-19** to compute content and style representations, with style transfer process optimized by minimizing content loss and style loss

Style Transfer – Model Comparison

Architecture of all models applied for model comparison

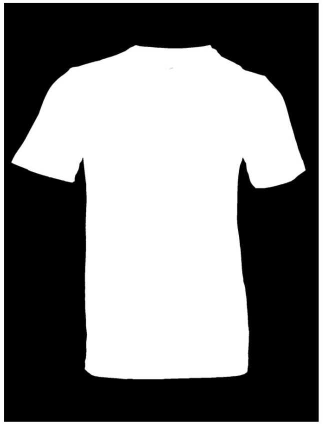
Model Name	**MobileNetV2-based feature extraction + Deep Learning-based Style Transfer**	**Hybrid Stylization Model with DeepLabV3-based segmentation**	**Hybrid Stylization Mode with UNet-based segmentation**	**Self-Trained Neural Style Transfer Model with VGG-19**
Strengths	Very fast, good generalization to other images	Selective stylization on the clothes only, natural blend between style and content, better boundary detection than UNet	More accurate **segmentation by pixels**, faster than DeepLabV3	**More customizable** and more control over training
Weaknesses	**Apply the style to entire image**, lacks fine control	sometimes **mislabel** object as background, processing time increases for high-resolution inputs	Poor generalization across different environmental conditions (e.g. lighting, poses, or textures)	**Extremely slow training**, may generate artifacts

Style Transfer – Model Comparison

Model performance measurement metrics

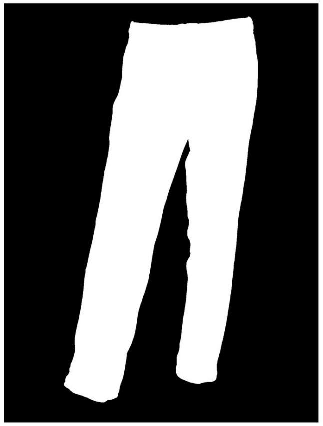
Original Image

Generated Ground Truth Mask



Original Image

Generated Ground Truth Mask



Model Name	DeepLabV3 + NST (sample_tshirt)	DeepLabV3 + NST (sample_pants)	UNet + NST (sample_tshirt)	UNet + NST (sample_pants)
IoU	0.971327	0.963041	0.011383	0.041226
Dice Score	0.632985	0.532510	0.483049	0.399213
Run Time	10.003325	13.243023	1.748159	1.758464

Style Transfer – Sample Application

Sample Style applied with Hybrid Stylization Mode with DeeplabV3-based segmentation

DeeplabV3-based Model - Sample Style#1

DeeplabV3-based Model - Sample Style#2

Potential Applications

OPTION

01

Enhancing **clothing customization** through AI-driven design

OPTION

02

Encouraging **artistic collaboration** between AI, designers, and clothing brands.

Future Steps and Business Impact

Highlight key strategies based on analysis and modelling result

Enhanced Product Discovery & User Experience – Improves product categorization, enabling seamless search and navigation for users, reducing friction in shopping experiences.
Style Transfer for Personalized Fashion – Enables AI-driven design customization, allowing users to create unique and artistic clothing by applying various visual styles.

01

Deploy the Best Prediction Model for E-commerce Operations
(*model deployment and operations plan in appendix 10)

- Convert best-performing prediction models into an API using Flask/FastAPI.
- Deploy the model to an online fashion retail platform for automatic product categorization.

02

Improved Product Discovery & Search Optimization

- Automatically categorizes products into **Apparel, Footwear, and Accessories**, enhancing search accuracy.
- Enables **faster product discovery**, reducing friction in the shopping experience.
- Supports **visual search** features, allowing users to find similar items by uploading an image.

03

Enhance Fashion Recommendation System

- Implement outfit-based recommendation using similarity analysis.
- Integrate personalized suggestions based on user preferences and previous purchases to boost engagement.

04

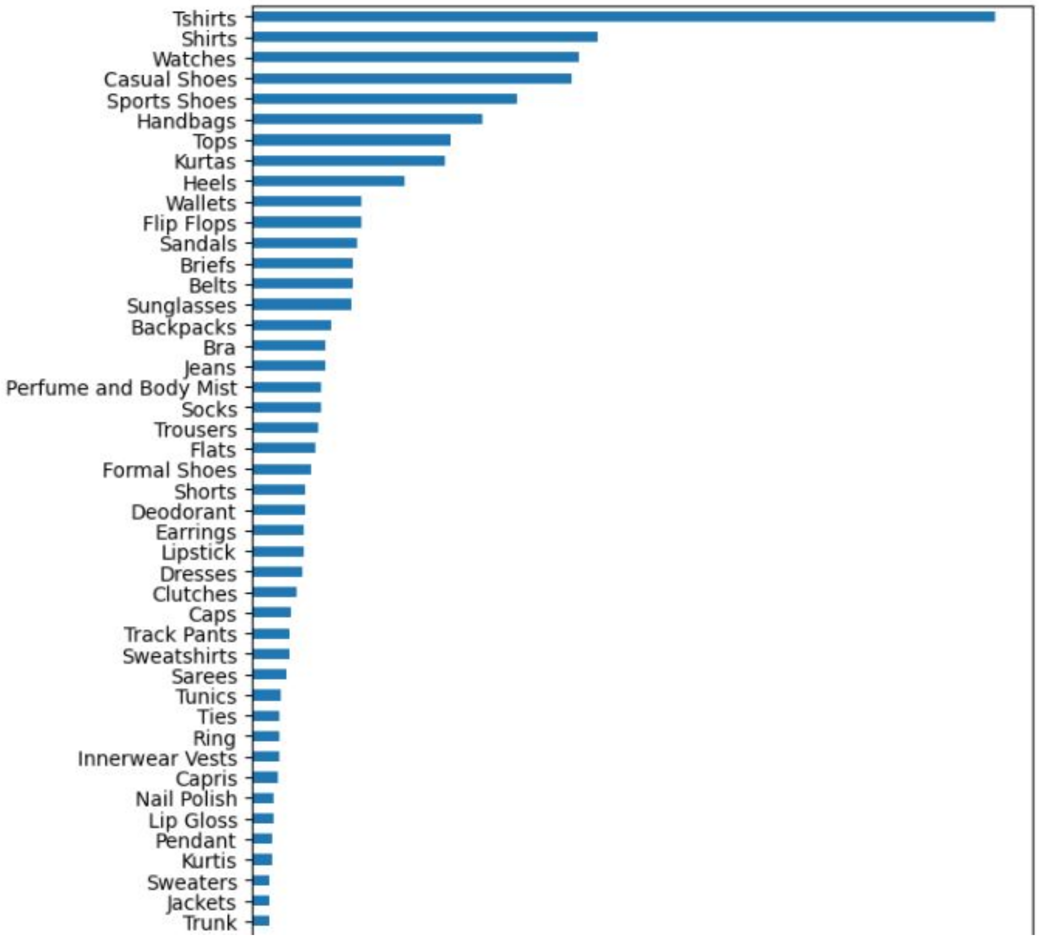
Expand Style Transfer for Virtual Try-On & Marketing

- Implement AI-powered style transfer for virtual outfit visualization.
- Enable customers to visualize clothing in different styles and colors, improving purchase confidence.

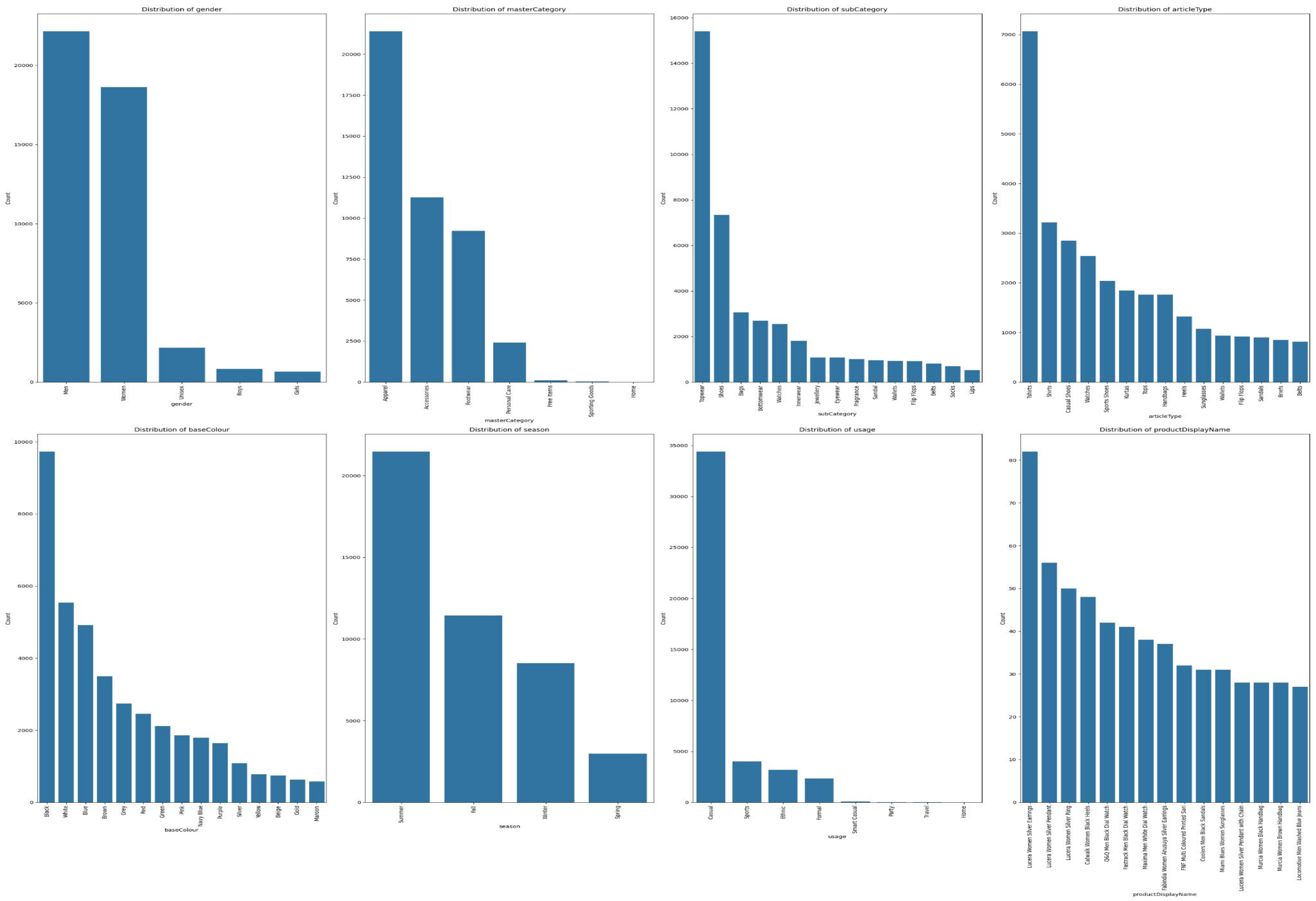
Thank you!

Appendix

Appendix 1: Distribution of clothing articles (part of the entire plot)



Appendix 2: distribution of all features individually



Appendix 3: Self-trained Classification Model Architecture

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 128, 128, 32)	896
max_pooling2d_9 (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_10 (Conv2D)	(None, 64, 64, 64)	18,496
max_pooling2d_10 (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_11 (Conv2D)	(None, 32, 32, 128)	73,856
max_pooling2d_11 (MaxPooling2D)	(None, 16, 16, 128)	0
flatten_3 (Flatten)	(None, 32768)	0
dense_10 (Dense)	(None, 1024)	33,555,456
dense_11 (Dense)	(None, 3)	3,075

Appendix 4: CNN-based feature extraction + Deep Learning-based Style Transfer

/content/drive/My Drive/Colab Notebooks/sample_tshirt.jpg /content/drive/My Drive/Colab Notebooks/sample_tshirt2.jpg /content/drive/My Drive/Colab Notebooks/sample_pants.jpg

Appendix 5: CNN-based feature extraction + Deep Learning-based Style Transfer

/content/drive/My Drive/Colab Notebooks/sample_tshirt.jpg

/content/drive/My Drive/Colab Notebooks/sample_tshirt2.jpg

/content/drive/My Drive/Colab Notebooks/sample_pants.jpg

Appendix 6: Hybrid Stylization Model with DeepLabV3-based segmentation

Appendix 7: Hybrid Stylization Model with DeepLabV3-based segmentation

ntent/drive/My Drive/Colab Notebooks/sample_tshirt2.jpg

content/drive/My Drive/Colab Notebooks/sample_pants.jpg

Appendix 8: Hybrid Stylization Mode with UNet-based segmentation

ɔntent/drive/My Drive/Colab Notebooks/sample_tshirt2.jpg

ɔntent/drive/My Drive/Colab Notebooks/sample_pants.jpg

Appendix 9: Hybrid Stylization Mode with UNet-based segmentation

Appendix 10: Classification Model Deployment and Operations