


# IN-SHOP CLOTHES RETRIEVAL ADJUSTED DEEPFASHION



## 第九組

B1228006 鄭晔薰 B1228007 張鴻毅 B1228022 梁釗豪  
B1228029 蘇琪文 B1228032 張紋菱

# TABLE OF CONTENT




01

任務介紹



03

資料前處理



02

EDA



04

預計方法

# 介紹

任務：服飾以圖搜圖

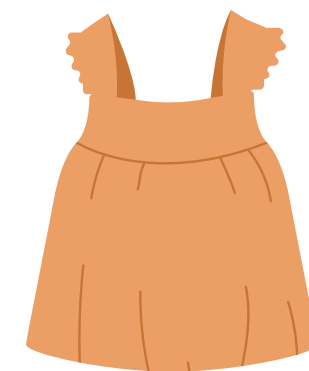
資料集特點：

- 影像內容：時尚服飾商品圖、模特兒穿搭照
- 標註形式：像素級別的多類別遮罩（16個身體/服飾部位）
- 挑戰難點：
  - 類別不平衡（背景佔大部分像素）
  - 細節豐富（蕾絲、紋理、刺繡）
  - 複雜邊界（衣物與背景、不同服飾之間）
  - 光照變化（攝影棚、戶外、不同時段）



## Content Name

Presentation is a public speaking activity that aims to convey a message or information.



## Content Name

Presentation is a public speaking activity that aims to convey a message or information.

# 資料集

## DeepFashion\_In-shop\_Clothes\_Retrieval\_Adjusted



### DeepFashion\_In-shop\_Clothes\_Retrieval\_Adjusted

Re-organized and Re-labeled Version of In-shop Clothes Retrieval Benchmark

 kaggle.com

<https://www.kaggle.com/datasets/hserdaraltan/deepfashion-inshop-clothes-retrieval-adjusted>



# 資料集

## 用途：

訓練與評估服裝檢索模型

(例如：給定一張服裝圖片，找出相似款式或同款  
不同角度的圖片)

## 來源：

根據原始 DeepFashion In-shop Clothes  
Retrieval Benchmark 調整而來

## 特色：

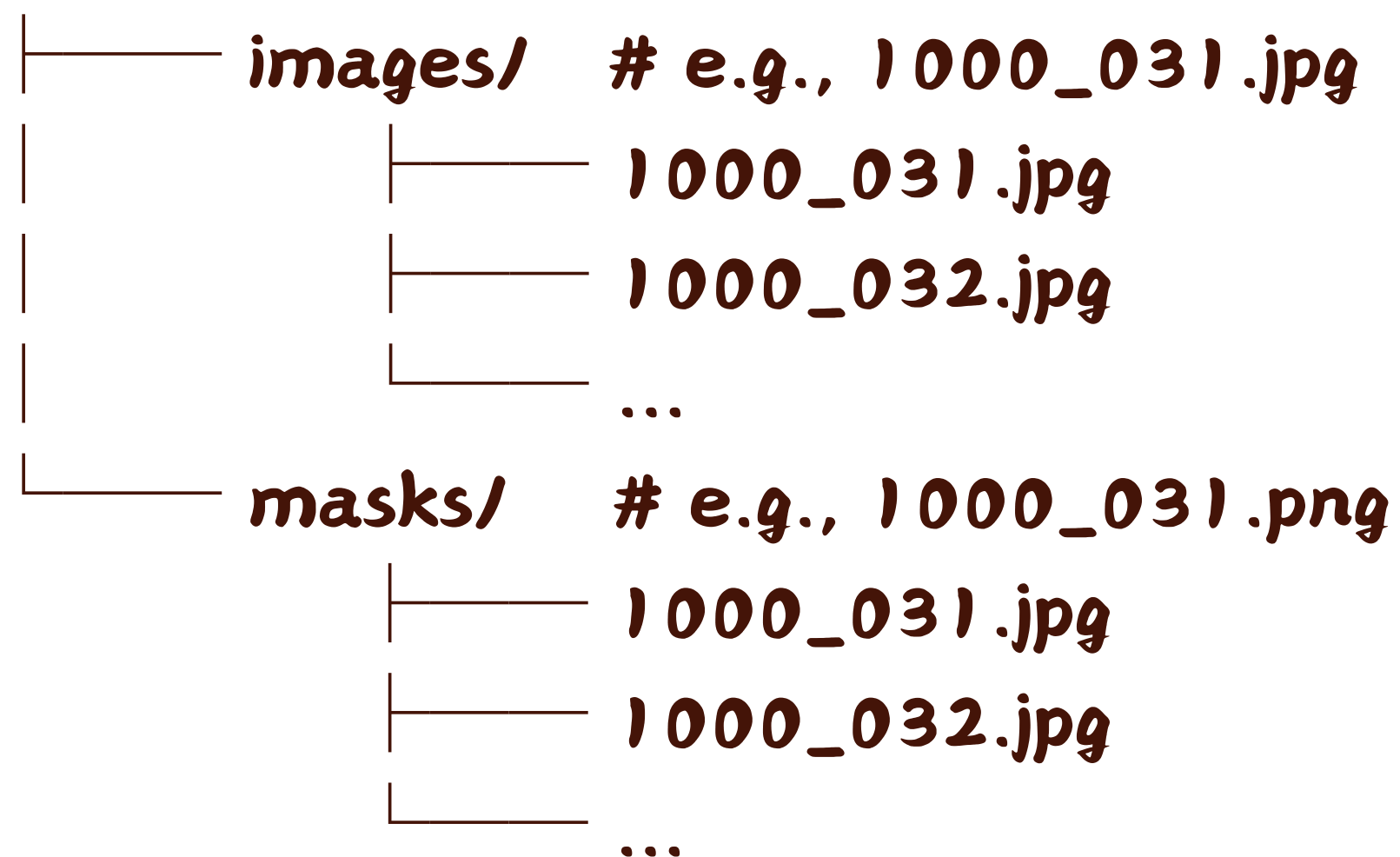
圖像已重新整理與標註



# EDA(資料分析)

## 資料結構

deepfashion/



## 取名結構

1000\_031.jpg



## 基本資訊

images/: 13752 files

masks/: 13752 files

# EDA(資料分析)

## 圖像樣本視覺化

1000\_031.jpg



1000\_032.jpg



1000\_041.jpg



1000\_042.jpg



1001\_011.jpg



1002\_011.jpg



Mask



Mask



Mask



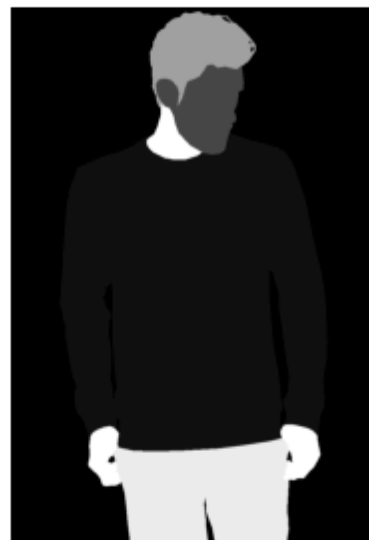
Mask



Mask



Mask



# EDA(資料分析)



## 圖像分布分析(商品)

商品總數：4600

圖片數量最多的前3個產品：

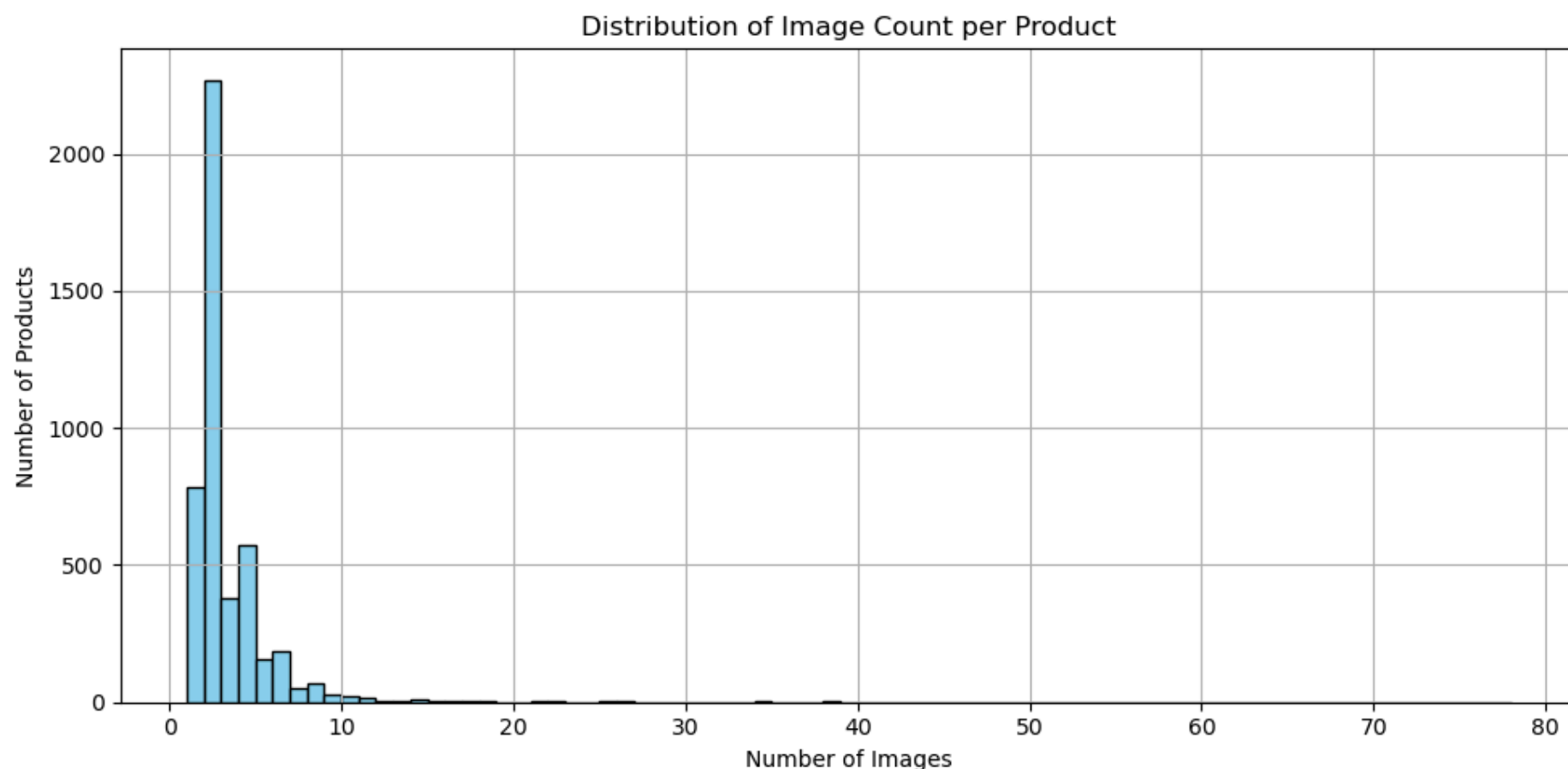
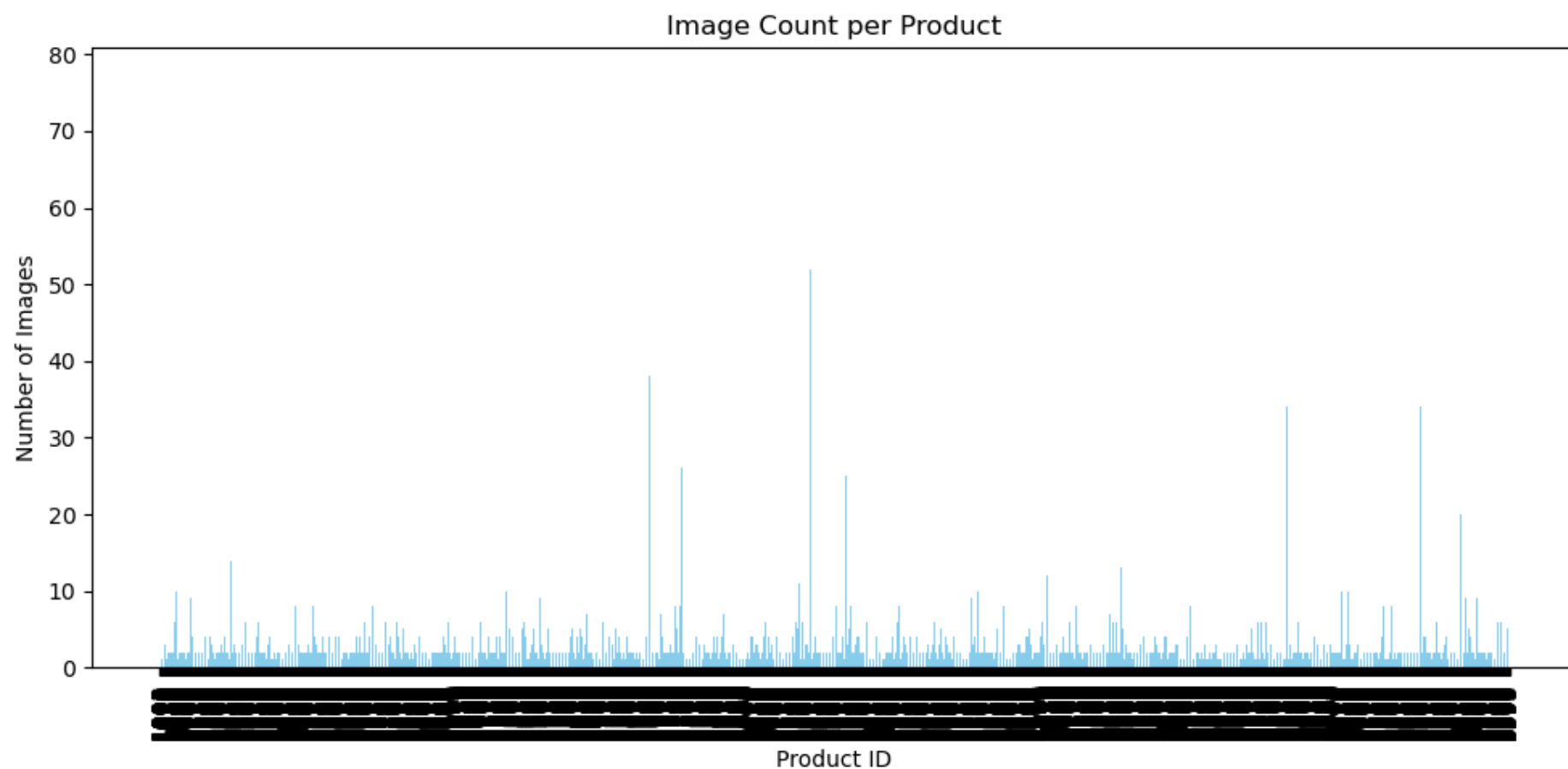
產品 1903：77 張

產品 2661：59 張

產品 2671：58 張

普遍的图片數量是 2 張，

共有 2270 個產品



# EDA(資料分析)

## 圖像分布分析(模特兒)

模特兒最多的前3個產品：

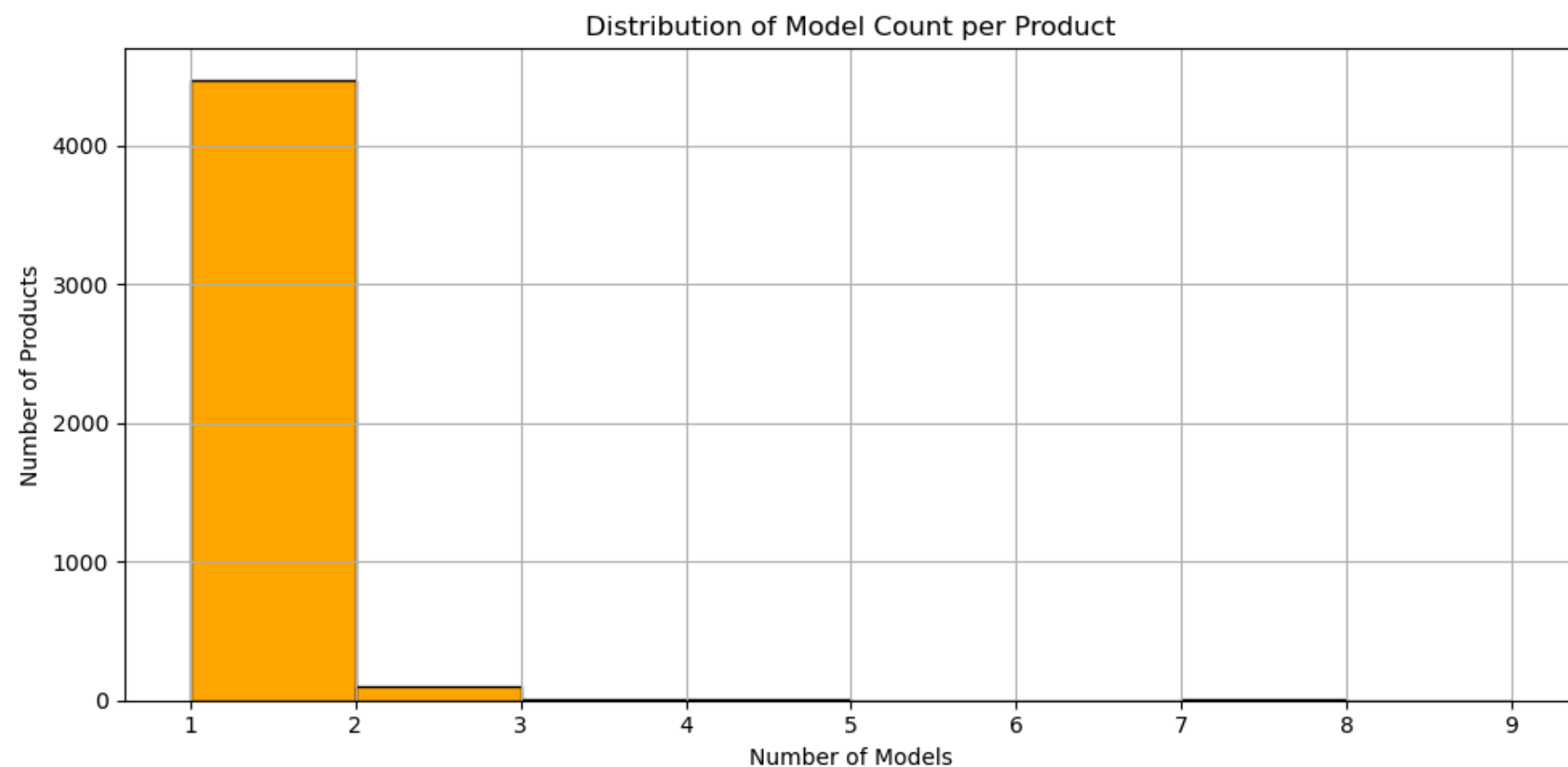
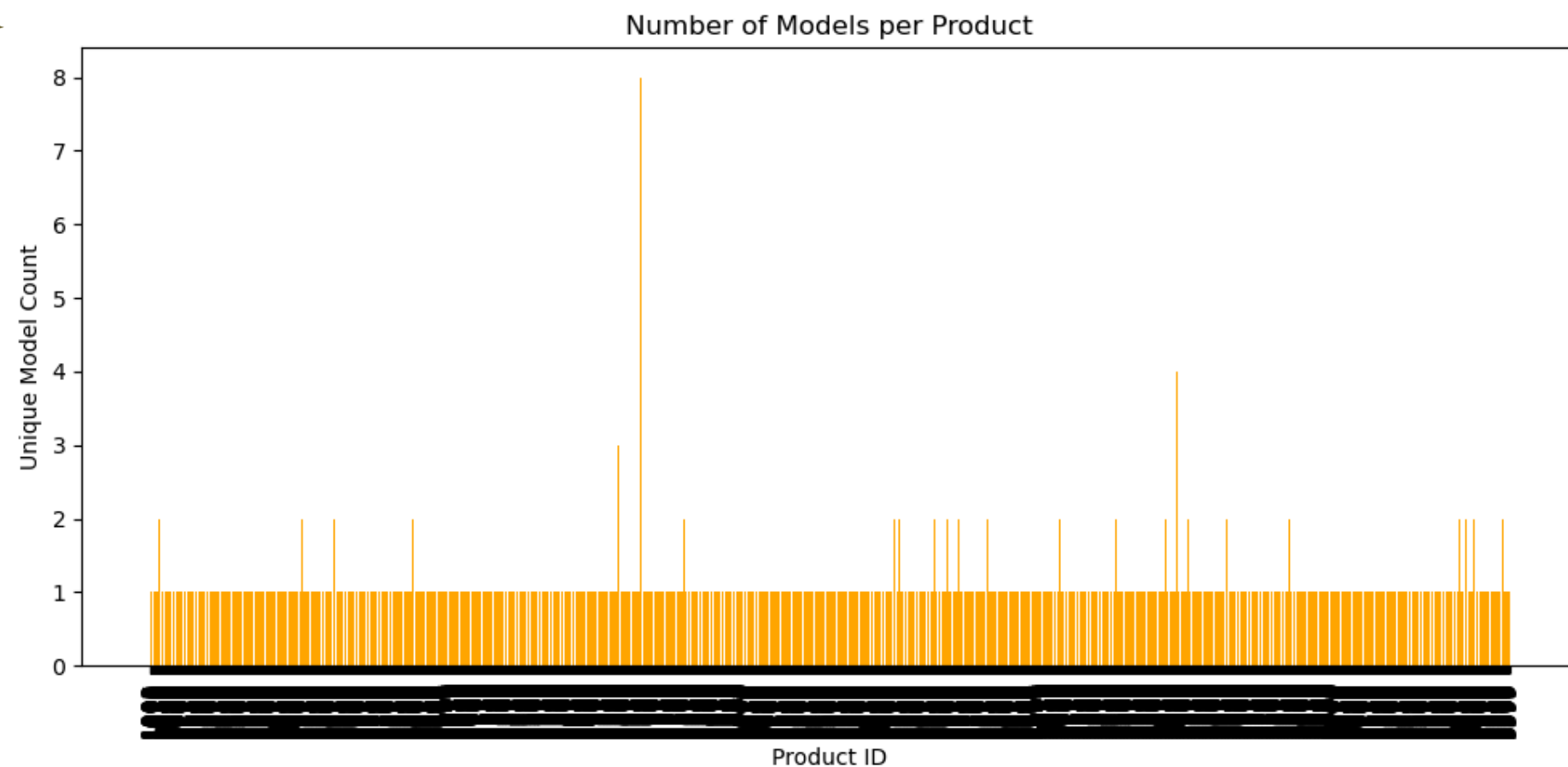
產品 1981：8 位模特兒

產品 2661：8 位模特兒

產品 1877：7 位模特兒

普遍的模特兒數量是 1 位，

共有 4475 個產品



# EDA(資料分析)



## 圖像分布分析(顏色)

顏色最多的前3個產品

(以最多模特兒為準)：

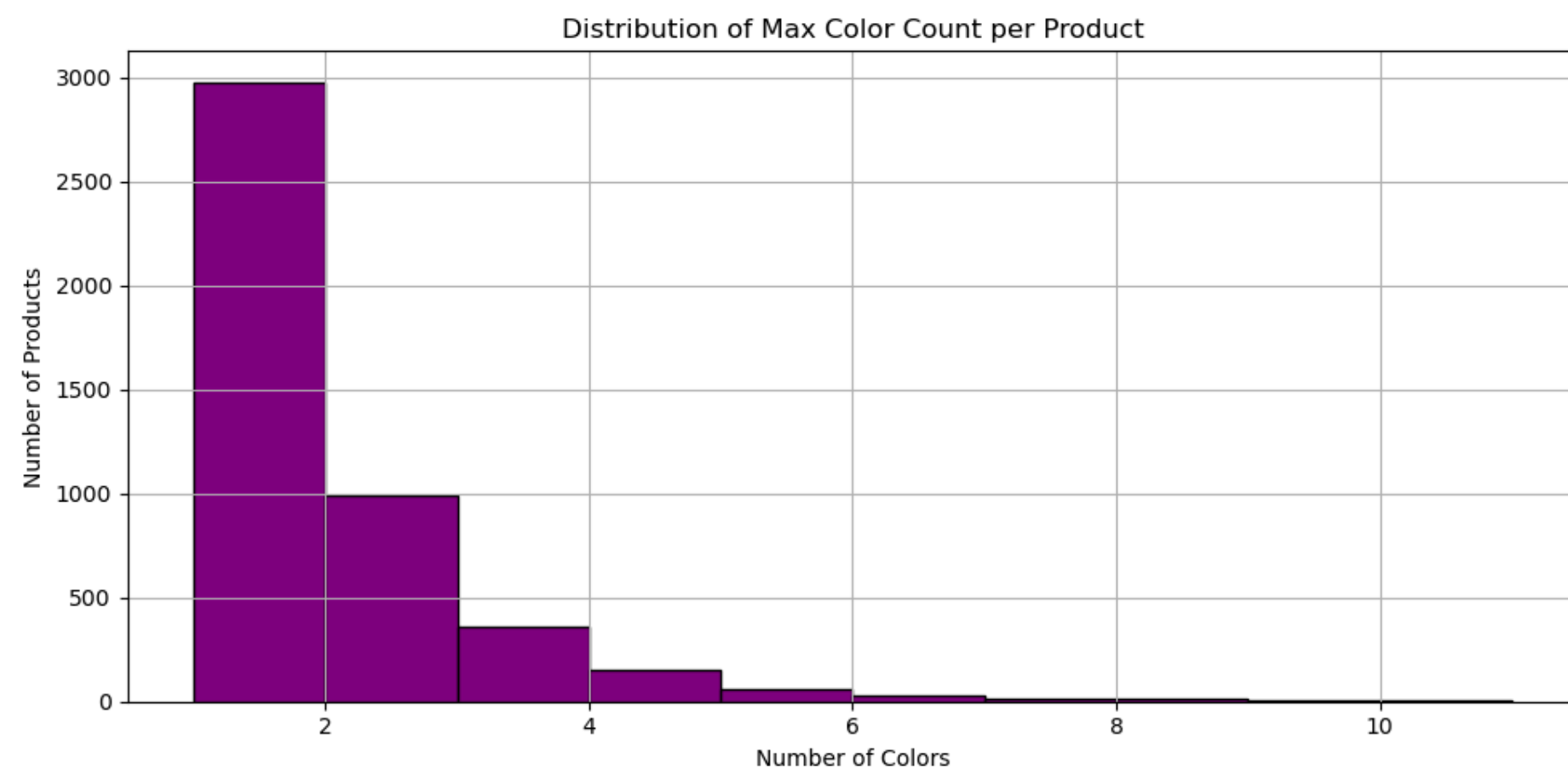
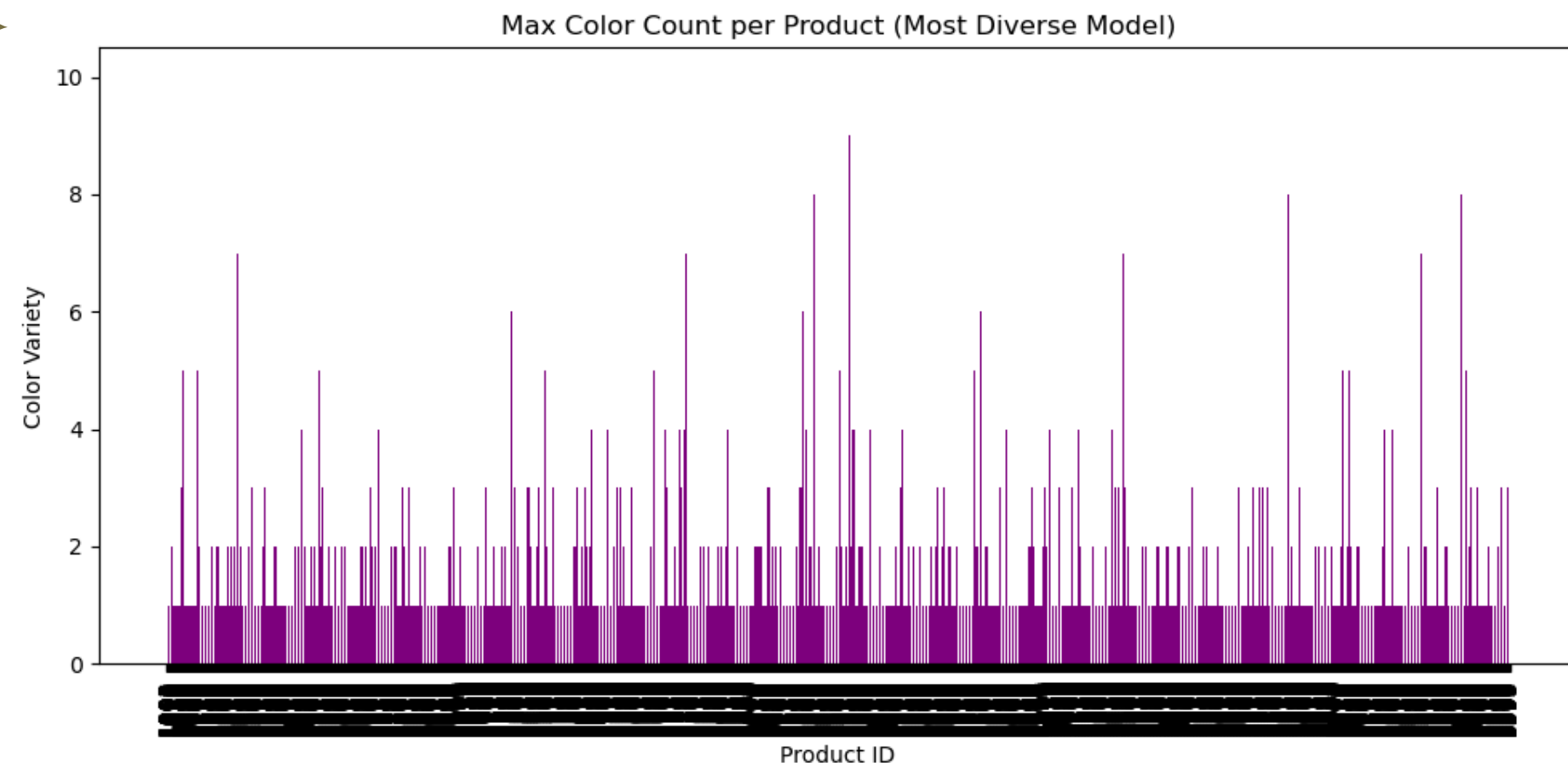
產品 5113：10 種顏色

產品 5239：10 種顏色

產品 2441：9 種顏色

普遍的顏色數量是 1 種，

共有 2978 個產品





Google Colab

google.com



# 資料前處理

## 1. 資料清洗與驗證

- 檢查檔案完整性
- 影像尺寸檢測
- 空白影像過濾

## 2. 資料分割

- `train/val/test`
- `item_id` 不跨 `split`
- 保持影像同步

## 3. 資料增強

- 調整亮度、飽和度、色相
- 隨機翻轉、旋轉、裁剪
- 類別權重計算 (加權交叉熵處理小物件)

## 4. 影像處理

- 解碼JPEG → 512x512縮放 → 正規化  
[-1,1]

## 5. 資料集建立

- 隨機抽樣
- 保持影像配對
- 固定隨機種子

# 資料前處理

## 1. 資料清洗與驗證

- 檢查檔案完整性
- 影像尺寸檢測
- 空白影像過濾

## 2. 資料分割

- `train/val/test`
- `item_id` 不跨 `split`
- 保持影像同步

📊 驗證結果：

=====

總數量： 13752

✅ 有效： 13752 (100.0%)

❌ 無效： 0 (0.0%)

🖼️ 顯示隨機樣本 (4張)：

/tmp/ipython-input-1042926275.py:75: UserWarning: Glyph 27171 (\N{CJK UNIFIED IDEOGRAPH-6A23}) missing from font(s) DejaVu Sans.  
plt.tight\_layout()

/tmp/ipython-input-1042926275.py:75: UserWarning: Glyph 26412 (\N{CJK UNIFIED IDEOGRAPH-672C}) missing from font(s) DejaVu Sans.  
plt.tight\_layout()

/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 27171 (\N{CJK UNIFIED IDEOGRAPH-6A23}) missing from font(s) DejaVu Sans.  
fig.canvas.print\_figure(bytes\_io, \*\*kw)

/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 26412 (\N{CJK UNIFIED IDEOGRAPH-672C}) missing from font(s) DejaVu Sans.  
fig.canvas.print\_figure(bytes\_io, \*\*kw)

🖼️ 1  
750x1101



🖼️ 2  
750x1101



🖼️ 3  
750x1101



🖼️ 4  
750x1101



# 資料前處理

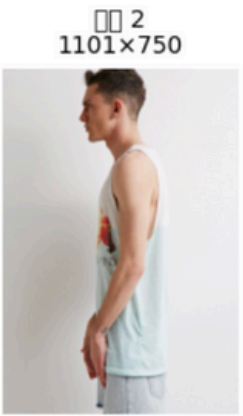
## 3. 資料增強

- 調整亮度、飽和度、色相
- 隨機翻轉、旋轉、裁剪
- 類別權重計算 (加權交叉熵處理小物件)



Item 1: 1912  
變體數量: 2  
樣本路徑: ['1912\_011.jpg', '1912\_012.jpg']

Item: 1912 (2 樣本)



Item 2: 1204  
變體數量: 2  
樣本路徑: ['1204\_011.jpg', '1204\_012.jpg']



隨機選擇 4 個樣本進行視覺化...

樣本 1:  
影像尺寸: (1101, 750, 3)  
遮罩尺寸: (1101, 750)  
遮罩唯一值: [ 0 1 4 7 9 13 14]  
影像像素範圍: [0, 255]  
遮罩像素範圍: [0, 14]

樣本 2:  
影像尺寸: (1101, 750, 3)  
遮罩尺寸: (1101, 750)  
遮罩唯一值: [ 0 1 13 14 15]  
影像像素範圍: [3, 255]  
遮罩像素範圍: [0, 15]

樣本 3:  
影像尺寸: (1101, 750, 3)  
遮罩尺寸: (1101, 750)  
遮罩唯一值: [ 0 1 4 7 8 9 13 14]  
影像像素範圍: [0, 255]  
遮罩像素範圍: [0, 14]

樣本 4:  
影像尺寸: (1101, 750, 3)  
遮罩尺寸: (1101, 750)  
遮罩唯一值: [ 0 1 4 7 9 13 14]  
影像像素範圍: [0, 255]  
遮罩像素範圍: [0, 14]

# 資料前處理



## 4. 影像處理

- 解碼JPEG → 512x512縮放 → 正規化[-1,1]

## 5. 資料集建立

- 隨機抽樣
- 保持影像配對
- 固定隨機種子

# COMPUTER VISION方法

## 特徵抽取

- **CNN(ResNet)**：用預訓練模型提取影像的特徵向量
  - 實作比較容易且穩定
  - 缺點是對細節敏感度有限
- **Vision Transformer**：是Transformer-based的模型
  - 擅長捕捉全局跟局部的細節
  - 準確率高
  - 缺點是模型比較大、訓練需要較長時間

# COMPUTER VISION方法

**Metric Learning**：為了讓相似的樣本更接近，不像的更遠

- **Siamese Network**：學習image之間的距離關係
  - 可以提升相似度的精確度
  - 缺點是需要樣本配對
- **ArcFace**
  - 穩定性高、收斂快、embedding效果好
  - 需要標籤分類

# COMPUTER VISION方法

## 特徵降維&可視化

- PCA：線性降維
  - 可保留主要變異
  - 加速檢索和可視化分群
  - 缺點是容易造成細節損失
- t-SNE：非線性降維
  - 可以保留局部結構，群集可視化效果不錯
  - 缺點是不太適合檢索

# COMPUTER VISION方法

## 相似度與檢索

- **Cosine Similarity**：一種衡量兩個向量之間「方向相似程度」的方法
  - 適合高維特徵
  - 可以跟FAISS、ArcFace結合使用
  - 缺點是會忽略幅度的差異、不適合非正規化的特徵

# COMPUTER VISION方法

## 相似度與檢索

- **FAISS**：Meta開發的高效向量相似度檢索庫
  - 適合用在以圖搜圖、影片片段搜尋還有商品/用戶推薦系統
  - 透過ANN搜尋、向量量化和GPU加速
  - 規模大效率高
  - 缺點是耗額外的記憶體去建索引、不支援動態頻繁更新

# COMPUTER VISION方法

資料強化：避免overfitting、強化資料多樣性

- 幾何變換

- 提升模型對視角變化的穩定性
- 缺點：訓練時間長

- 色彩增強

- 增強模型對光線/顏色的適應性
- 缺點：有可能會造成影像不自然



# COMPUTER VISION方法

## 分群、優化

- **K-means**
  - 把相似衣服群組起來
  - 可以快速篩群
- **Re-ranking(用在檢索之後)**
  - 對於檢索結果進行再排序
  - 可以提升精度
  - 缺點：需要額外運算

# 已有的 CODE 分析

參考來源：<https://github.com/open-mmlab/mmfashion>

## 使用其檢索 (Retrieval) 模塊

它支援將圖像轉成 embedding、然後用檢索流程 (query → gallery) 進行比對

## Backbone + 特徵抽取流程

MMFashion 的架構可選用常見 backbone (如 ResNet50) + 抽取 embedding，兼容替換 backbone，因此可從預訓練模型出發，再針對資料集做微調

## 資料格式與評估支援

MMFashion 為 In-shop / Consumer-to-Shop 檢索任務定義了資料格式、評估指標 (如 top-k recall) 及測試腳本

# 參考文獻

- DeepFashion2 Dataset Paper – “DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images.”
- <https://mmlab.ie.cuhk.edu.hk/projects/DeepFashion/InShopRetrieval.html>
- <https://www.kaggle.com/datasets/hserdaraltan/deepfashion-inshop-clothes-retrieval-adjusted>
- <https://github.com/facebookresearch/faiss>
- <https://ai.meta.com/tools/faiss/>
- Day 8 – Cosine Similarity：你我之間的夾角
- <https://blog.csdn.net/q923714892/article/details/119838895>
- <https://www.oryoy.com/news/jie-mi-arcface-shen-du-xue-xi-zhong-de-tu-xiang-shi-bie-li-qi-xiang-jie-qi-yuan-li-yu-shi-ji-ying-yo.html>

**THANK  
YOU**

