Classifying Fashion Images with Machine Learning

Shadi Farzankia	107209
Shruti Pashine	106369
Hemant	106007
Dharampal Singh	106316

https://github.com/AadrianLeo/Fashion-Style-Classifier

Main Goal

Classify Fashion related dataset into 5 Categories:

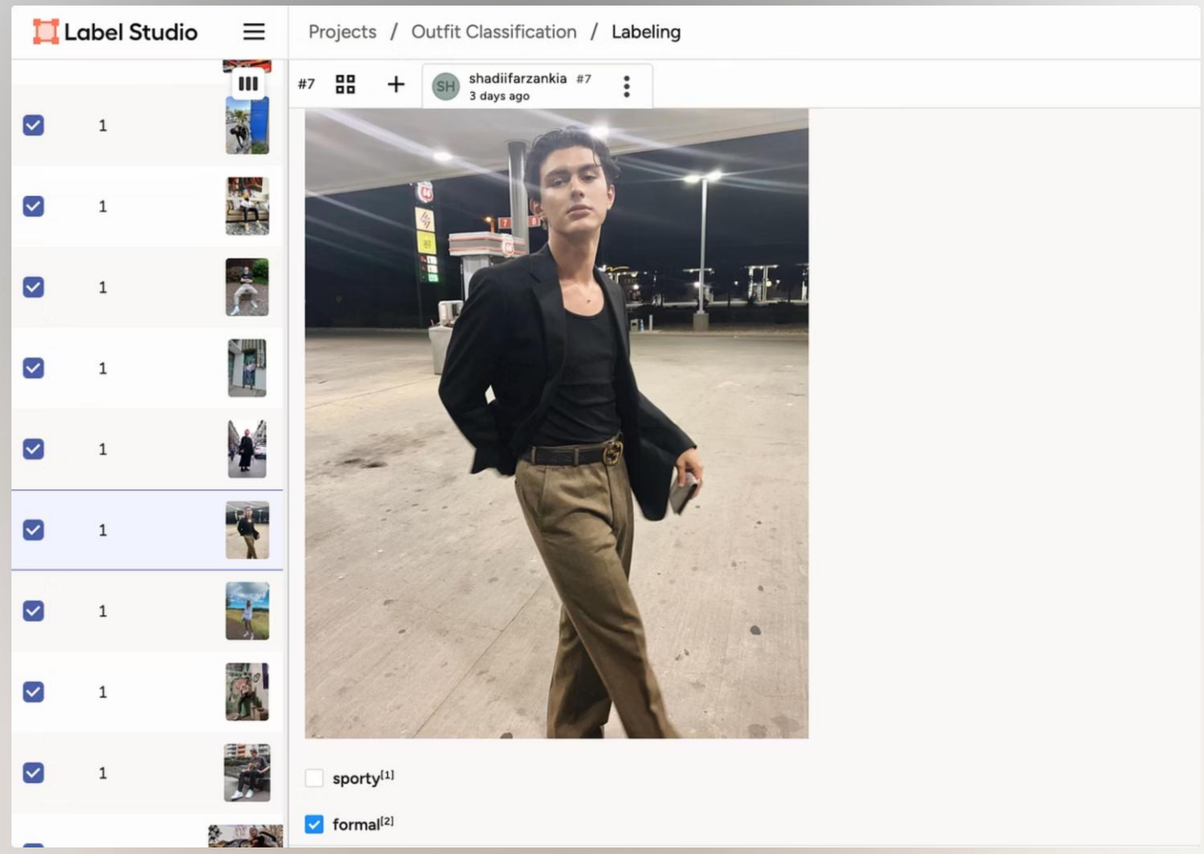
- StreetWear
- Formal
- Vintage
- Sporty
- Casual

Motivation

- Understanding fashion styles helps power recommendation engines, trend analysis, and personalized styling.

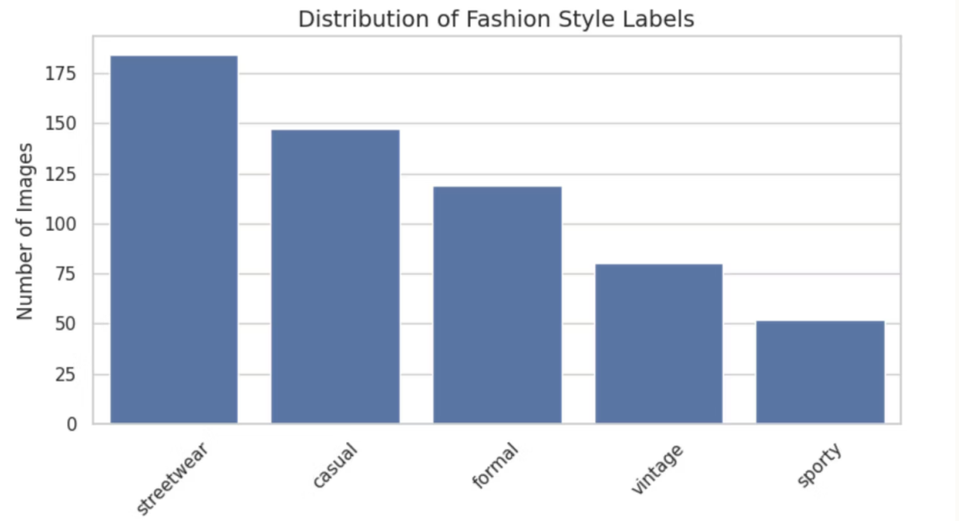
Dataset Details

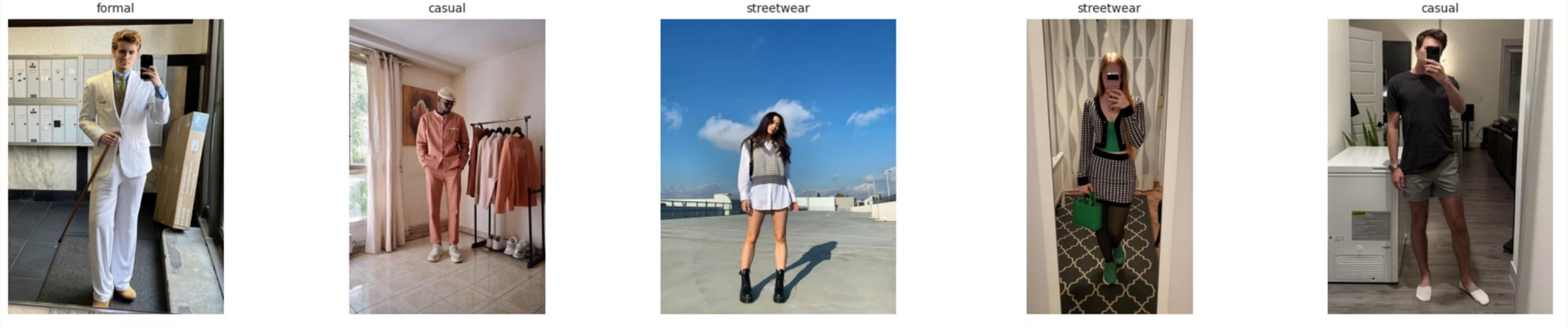
- Used Reddit **PRAW API** to Obtain Fashion-Related Subreddits
- Used hashtags such as **#streetwear**, **#fashion**, and **#femalefashionadvice**
- Relevant images were filtered and labeled into 5 style categories using **Label Studio**
- Removed **non-fashion content**, such as:
  - Tweet screenshots
  - Memes or text-only posts
  - Filtered out images with **multiple people** to maintain style consistency



Distribution of the Data

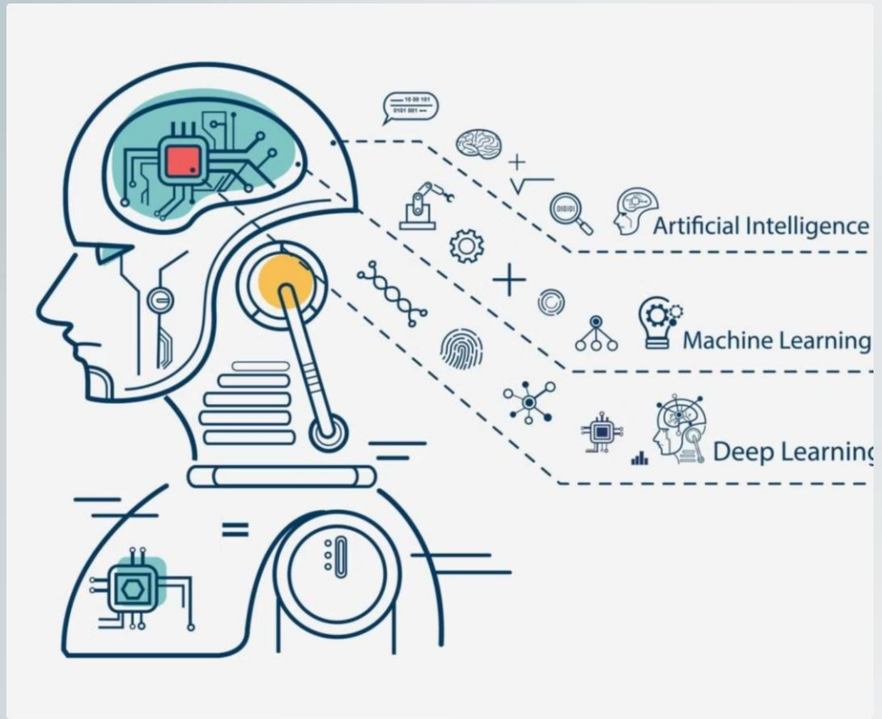
Category	Percentage
Streetwear	31.9%
Casual	25.0%
Formal	20.7%
Vintage	13.8%
Sporty	8.6%





Models Used

1. **Naive Baseline**
2. **K-Nearest Neighbors (KNN)**
3. **Neural Network (EfficientNet-B0)**



1. Naive BaseLine

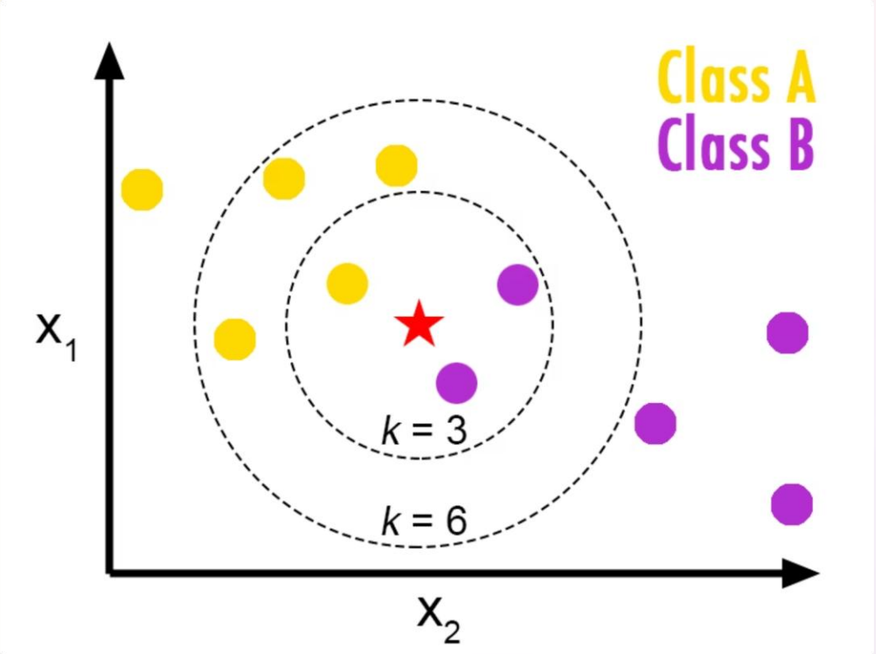
- predicts the most common class
- minimum expected performance
- For checking if our models learn anything at all

Naive Baseline Accuracy: 0.3162393162393162

	precision	recall	f1-score	support
casual	0.00	0.00	0.00	30
formal	0.00	0.00	0.00	24
sporty	0.00	0.00	0.00	10
streetwear	0.32	1.00	0.48	37
vintage	0.00	0.00	0.00	16
accuracy			0.32	117
macro avg	0.06	0.20	0.10	117
weighted avg	0.10	0.32	0.15	117

2. K-Nearest Neighbours

- a non-parametric algorithm
- classifies a sample based on the majority label of its closest training examples in the feature space
- pipeline:
  - **Feature Extraction -** Extracted deep image features using a pre-trained **MobileNetV2** model (without the classification head)
    - **MobileNetV2:** Pretrained on ImageNet
  - **Dimensionality Reduction -** Applied **PCA** to reduce the feature space and improve computational efficiency
    - Keep enough principal components to explain 95% of the variance in the data
  - **Feature Standardization -** Used **StandardScaler** to normalize features for distance-based comparison.
  - **Hyperparameter Tuning -** Tuned `k` and distance metrics using **GridSearchCV** with **5-fold cross-validation**.
  - Train/Validation Split: Stratified split to maintain class balance.



Results for KNN:

- **Overall Accuracy**: 41% — indicates moderate performance
- **Best Classified**:
  - **Formal** → F1-score: 0.55
  - **Streetwear** → F1-score: 0.56
- **Worst Classified**:
  - **Sporty** → F1-score: 0.00
- **Macro Avg. F1-score**: 0.29
  - Reflects poor performance on minority classes

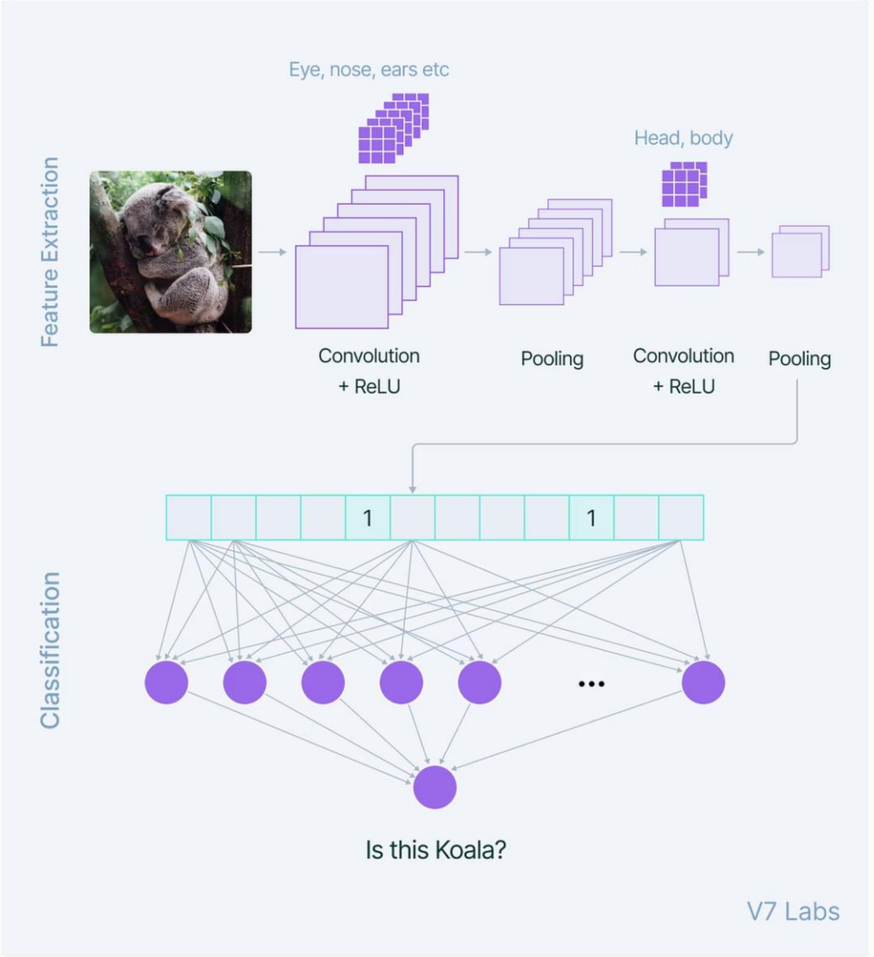
```
Accuracy: 0.41025641025641024
Classification Report:
              precision    recall  f1-score   support

      casual       0.25      0.13      0.17        30
      formal       0.43      0.75      0.55        24
      sporty       0.00      0.00      0.00        10
  streetwear       0.50      0.65      0.56        37
     vintage       0.20      0.12      0.15        16

    accuracy                           0.41       117
   macro avg       0.28      0.33      0.29       117
weighted avg       0.34      0.41      0.36       117
```

Convolutional Neural Network (CNN)

- **Model:** EfficientNet-B0 (pretrained on ImageNet, fine-tuned on fashion dataset)
- **Pipeline:**
  - **Data Preprocessing**: Resized images, applied augmentation (flip, rotation, color jitter), normalized pixel values.
  - **Train/Validation Split**: Stratified split to maintain class balance.
  - **Model Architecture**: **EfficientNet-B0** with custom classification head.
  - **Training**: Cross-entropy loss, Adam optimizer, trained for 20 epochs.
  - **Evaluation**: Classification report (accuracy, precision, recall, F1-score).



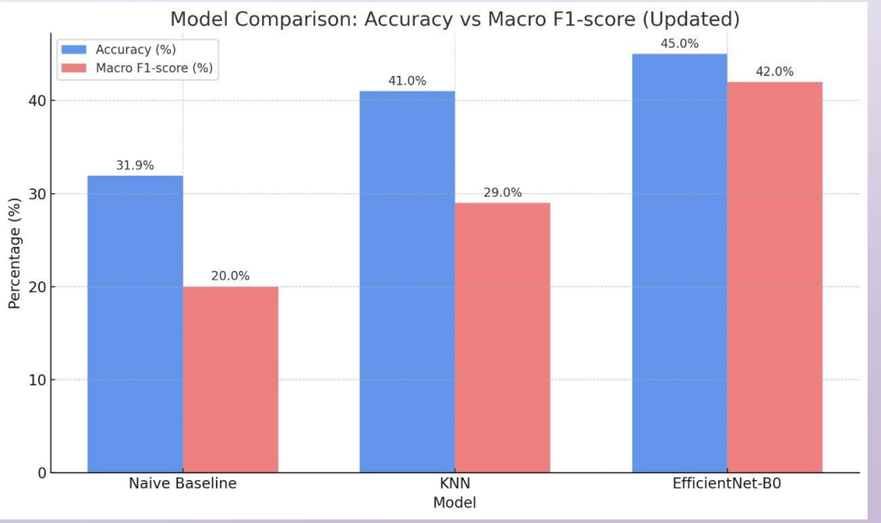
CNN Results (EfficientNet-B0):

- Overall Accuracy: 45%
- Best Classified:
  - Formal → F1-score: 0.64
  - Streetwear → F1-score: 0.5
- Worst Classified:
  - Sporty → F1-score: 0.27
- Macro Avg. F1-score: 0.42

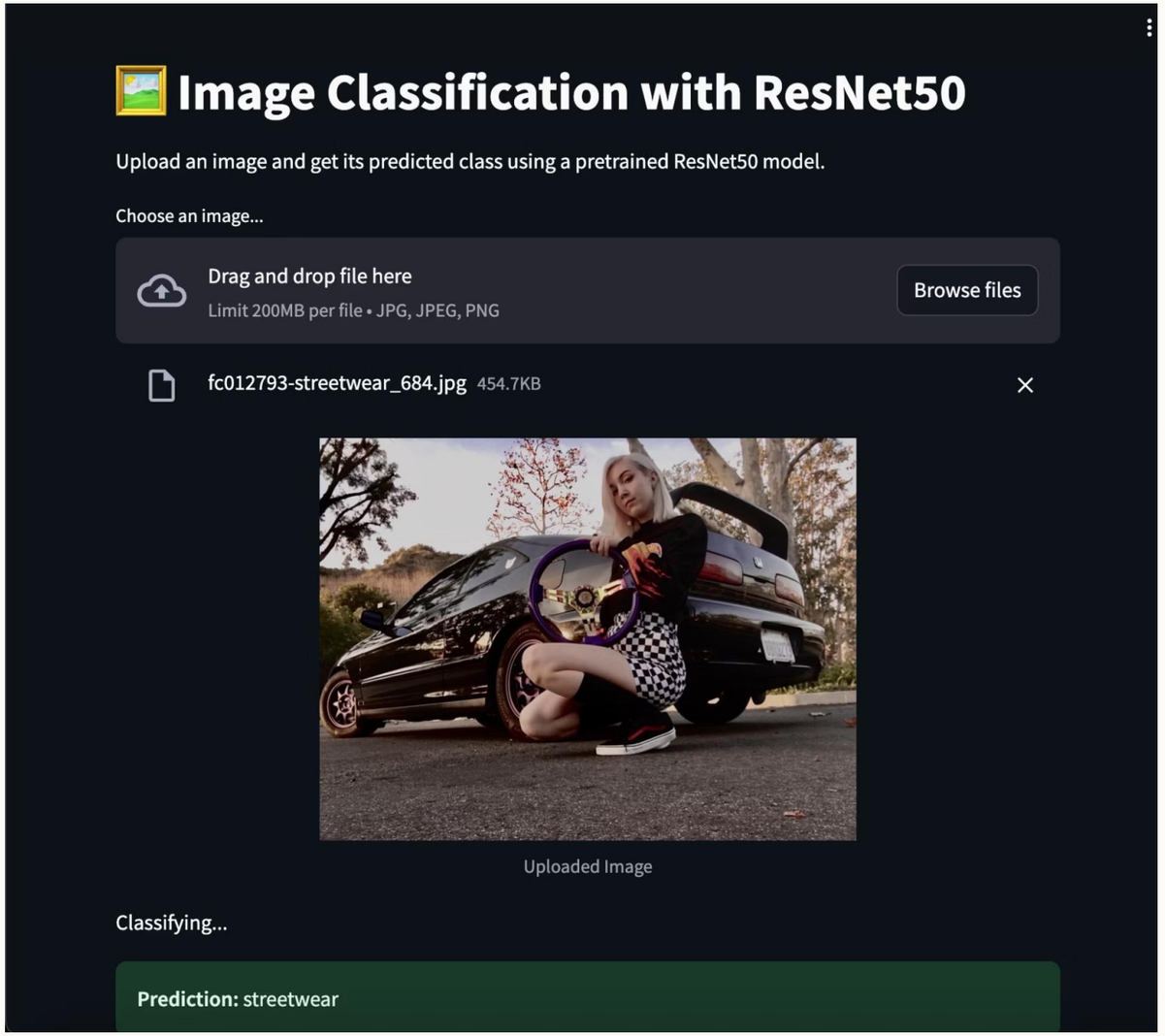
	precision	recall	f1-score	support
casual	0.42	0.37	0.39	30
formal	0.62	0.67	0.64	24
sporty	0.25	0.30	0.27	10
streetwear	0.51	0.49	0.50	37
vintage	0.28	0.31	0.29	16
accuracy			0.45	117
macro avg	0.42	0.43	0.42	117
weighted avg	0.46	0.45	0.45	117

Results Comparison:

Metric	Naive Baseline	KNN	EfficientNet-B0
Accuracy	31.9%	41%	**%45**
Macro F1-Score	~0.20	0.29	**0.42**
Best Classified	Streetwear	Streetwear/Formal	**Formal/Streetwear**
Worst Classified	All but most common	Sporty	Sporty (still low)



Streamlit App: Predicting Fashion Styles from Images



Future Work:

- **Expert guidance** for more accurate and consistent image labeling.
- **Apply stronger data augmentation** techniques to help balance underrepresented classes.
- **Use oversampling or generate synthetic images** (e.g., with GANs) to improve performance on minority categories.
- **Expand the dataset** with more diverse and balanced fashion images to enhance model generalization.
- **Tune hyperparameters** like learning rate, batch size, and number of epochs to optimize each model's performance.
- **Add explainability tools** such as Grad-CAM to better understand and visualize how the model makes decisions.