Name: Shubham Madhukar Patil

Fashion Recommendation System using Image Features

Introduction

In the era of online shopping, people are often overwhelmed by the variety of fashion products available on e-commerce websites. Searching through hundreds of dresses, tops, or accessories to find something visually similar to what one likes can be time-consuming and frustrating. To overcome this challenge, artificial intelligence and computer vision techniques can be used to build an intelligent **fashion recommendation system** that suggests visually similar outfits to users.

This project focuses on building such a recommendation system using **image feature extraction through deep learning**. The system uses a pre-trained convolutional neural network (CNN) model called **VGG16**, which learns visual patterns from images and extracts meaningful representations. These extracted features are then compared using similarity measures, enabling the model to recommend fashion items that look similar to the input image.

Unlike traditional recommender systems that rely on textual data such as tags, descriptions, or user ratings, this system is purely **content-based** and depends only on the **visual appearance of clothing**. This approach makes the system more general and suitable for situations where product descriptions are incomplete or unavailable.

Objective

The main objective of this project is to create a system that can recommend visually similar fashion items based on a reference image provided by the user. Specifically, the goals are:

1. To extract visual features from images using a pre-trained CNN model (VGG16).
2. To compute similarities between fashion images based on their extracted features.
3. To recommend and display the top-N images that are most visually similar to the input image.
4. To demonstrate how computer vision can be used in e-commerce and fashion analytics to enhance user experience.

Dataset Description

The dataset used in this project contains around **97 images** of women’s fashion items, including dresses, sarees, skirts, and tops. The dataset was provided in a compressed format named **‘women fashion.zip’**, which was extracted in the Google Colab environment. After extraction, the main directory contained a folder named *'women fashion'* and a metadata folder *'__MACOSX'* generated by macOS. The metadata folder was ignored.

Each image file has a clear background and good lighting conditions, making it suitable for visual analysis. The images were saved in standard formats such as `.jpg`, `.png`, and `.jpeg`. The `glob` module was used to list all image file paths from the folder, ensuring only valid image files were included.

A sample image was displayed using the `PIL` and `matplotlib` libraries to confirm that the images loaded correctly and to understand their resolution and color characteristics. This step also helped verify that all images were visually suitable for deep learning processing.

Methodology

1. Data Preparation and Extraction

The dataset was first extracted using Python's `ZipFile` library. The directory structure was checked and created if not already present using `os.makedirs()`. Once extracted, the images were listed and inspected. The `glob` module was used to automatically identify all image files from the specified directory.

This data preparation ensured that the model worked only on valid image paths, which prevented any file-related errors during feature extraction.

2. Image Feature Extraction using VGG16

Feature extraction is the most critical step in this system. For this, the **VGG16 model** was used, which is a 16-layer deep convolutional neural network trained on the ImageNet dataset. Since VGG16 has already learned to identify a variety of visual features such as textures, edges, and shapes, it can be repurposed for other image-related tasks without needing retraining.

In this project, the **top classification layer** of VGG16 was removed (`include_top=False`), allowing it to function purely as a **feature extractor**. Each image was resized to 224×224 pixels, converted to a NumPy array, and preprocessed using the `preprocess_input()` function to match the format expected by the model.

Once passed through the model, each image produced a **feature map**, which was then flattened and normalized to form a fixed-length feature vector. These vectors represent the essential visual characteristics of each fashion image in numerical form.

3. Measuring Similarity

After obtaining the feature vectors, similarity between images was calculated using the **cosine similarity** metric. Cosine similarity measures how closely two vectors point in the same direction, making it suitable for comparing feature representations.

The similarity between the input image's feature vector and all other vectors in the dataset was computed, and the most similar images were identified based on the highest similarity scores.

4. Recommendation Function and Visualization

A custom function named `recommend_fashion_items_cnn()` was implemented to display recommendations. This function performs the following steps:

1. Takes an input image path from the dataset.
2. Extracts its features using the VGG16 model.
3. Computes cosine similarity between the input image and all dataset images.
4. Sorts the similarity scores in descending order.
5. Displays the input image alongside the top-N most similar fashion items.

Using `matplotlib`, the system visually presents the input image and the recommended ones in a single figure. This visualization clearly shows how the recommendations resemble the input in terms of color, design, and pattern.

Results

The fashion recommendation system successfully identified and displayed similar images based on visual appearance. For instance, when an image of a black sleeveless dress was used as input, the system recommended other dark-colored dresses with similar textures and designs.

This confirms that the **VGG16 model captures the underlying visual features effectively**. The recommendations appeared logical and visually relevant. Each recommendation shared similar style elements, such as neckline shape, sleeve type, or color tone.

Although the dataset was relatively small (97 images), the system performed consistently well, demonstrating the potential of deep learning–based feature extraction in building intelligent image recommendation systems.

Discussion

This project highlights how deep learning can be applied in practical scenarios beyond classification. The approach is **unsupervised** — no labels or categories were required — and yet, the system could understand and recommend visually related fashion items.

One limitation is that this system depends entirely on visual similarity. It does not consider other attributes such as brand, material, or user preferences. In real-world e-commerce systems, combining visual features with text-based attributes and user behavior data could lead to even better and more personalized results.

Despite these limitations, the project demonstrates how easily pre-trained CNN models can be reused to solve new problems with limited data and resources.

Future Scope and Enhancements

There are several ways this project can be extended:

- **Use of advanced models:** Models like **ResNet50**, **InceptionV3**, or **EfficientNet** could provide richer and more discriminative feature representations.
- **Efficient similarity search:** For larger datasets, using libraries such as **FAISS (Facebook AI Similarity Search)** or **Annoy** would make similarity computation faster.
- **Web application deployment:** The system can be deployed using **Streamlit** or **Flask**, allowing users to upload an image and instantly receive recommendations.
- **Hybrid recommendation:** Combining image features with text features (like color, fabric, or price) could create a **hybrid recommendation system**, improving accuracy and user satisfaction.
- **User personalization:** Future versions can learn from user choices to provide personalized fashion suggestions.

Conclusion

The Fashion Recommendation System using Image Features demonstrates how deep learning and computer vision can transform the way fashion items are discovered online. By relying solely on visual features, the system bypasses the need for manual tagging and textual data.

Through this project, I learned how to extract deep features from images using **transfer learning** and how to compare those features to identify visual similarity. The results were promising and show clear potential for application in e-commerce platforms.

In summary, this system bridges the gap between artificial intelligence and fashion, making online shopping more engaging, intelligent, and efficient.