# FASHION RECOMMENDATION SYSTEM USING IMAGE SIMILARITY

PROJECT REPORT

Submitted by:  
K JEYA BHARATHI,

C HEMAPATHI,

J LEENA CHELSIN.

Department of Computer Applications  
Madurai Kamaraj University

## TABLE OF CONTENT

| **1.** | **1. ABSTRACT** |
| --- | --- |
| **2.** | **2. PROBLEM DEFINITION & DESCRIPTION** |
| **3.** | **3. SYSTEM ENVIRONMENT** |
| **4.** | **3.1 HARDWARE REQUIREMENT** |
| **5.** | **3.2 SOFTWARE REQUIREMENT** |
| **6.** | **4. SYSTEM STUDY AND ANALYSIS** |
| **7.** | **4.1 EXISTING SYSTEM** |
| **8.** | **4.2 PROPOSED SYSTEM** |
| **9.** | **5. SYSTEM DESIGN** |
| **10.** | **6. SYSTEM TESTING** |
| **11.** | **7. MODULES** |
| **12.** | **8. OUTPUT DESIGN** |
| **13.** | **9. PROJECT CODE** |
| **14.** | **10. CONCLUSION** |
| **15.** | **11. FUTURE ENHANCEMENT** |

## 

## CHAPTER – 1 ABSTRACT

The Fashion Recommendation System using Image Similarity is an AI-based project designed to find and suggest visually similar clothing or fashion items. The system uses deep learning techniques and a pre-trained model, MobileNetV2, to extract image features and compare them using cosine similarity. This allows users to discover related fashion products easily and efficiently. It is particularly useful in online shopping platforms to enhance user experience and increase engagement.

## CHAPTER – 2 PROBLEM DEFINITION & DESCRIPTION

In today’s digital marketplace, customers often find it difficult to locate visually similar fashion products based on appearance. Traditional search methods rely on text-based queries, which do not effectively capture visual features like color, texture, and design. This project aims to solve this issue by developing a system that recommends fashion items based on image similarity, enhancing the shopping experience and reducing manual browsing.

## CHAPTER – 3 SYSTEM ENVIRONMENT

### 3.1 HARDWARE REQUIREMENT

Processor : Intel(R) Core i3 or above  
RAM : 4 GB or higher  
Hard Disk : Minimum 500 GB  
System Type : 64-bit Operating System

### 3.2 SOFTWARE REQUIREMENT

Operating System : Windows   
Programming Language : Python IDE  
Libraries : TensorFlow, Keras, NumPy, OpenCV, Matplotlib, scikit-learn, tqdm  
IDE : Python

## CHAPTER – 4 SYSTEM STUDY AND ANALYSIS

### 4.1 EXISTING SYSTEM

Existing fashion recommendation systems mostly rely on text-based search and manual tagging, which are time-consuming and inaccurate. They fail to identify visual similarities between fashion items and do not provide an intuitive search experience.

### DISADVANTAGES

* Difficult to find visually similar items.
* Relies heavily on text-based metadata.
* Cannot detect visual patterns, colors, or styles.
* Low accuracy in personalized recommendations.

### 4.2 PROPOSED SYSTEM

The proposed system uses a pre-trained MobileNetV2 model to extract image features and compute visual similarity between items. The model efficiently represents visual data, allowing the system to find and recommend similar images based on deep learning feature extraction.

### ADVANTAGES

* Accurate and automated image similarity detection.
* Fast processing using pre-trained models.
* Enhances user shopping experience.
* Lightweight and runs on laptops efficiently.

## CHAPTER – 5 SYSTEM DESIGN

The system design includes loading the dataset, extracting features using MobileNetV2, and comparing them using cosine similarity. The design also integrates visualization using matplotlib to display similar fashion items.

## CHAPTER – 6 SYSTEM TESTING

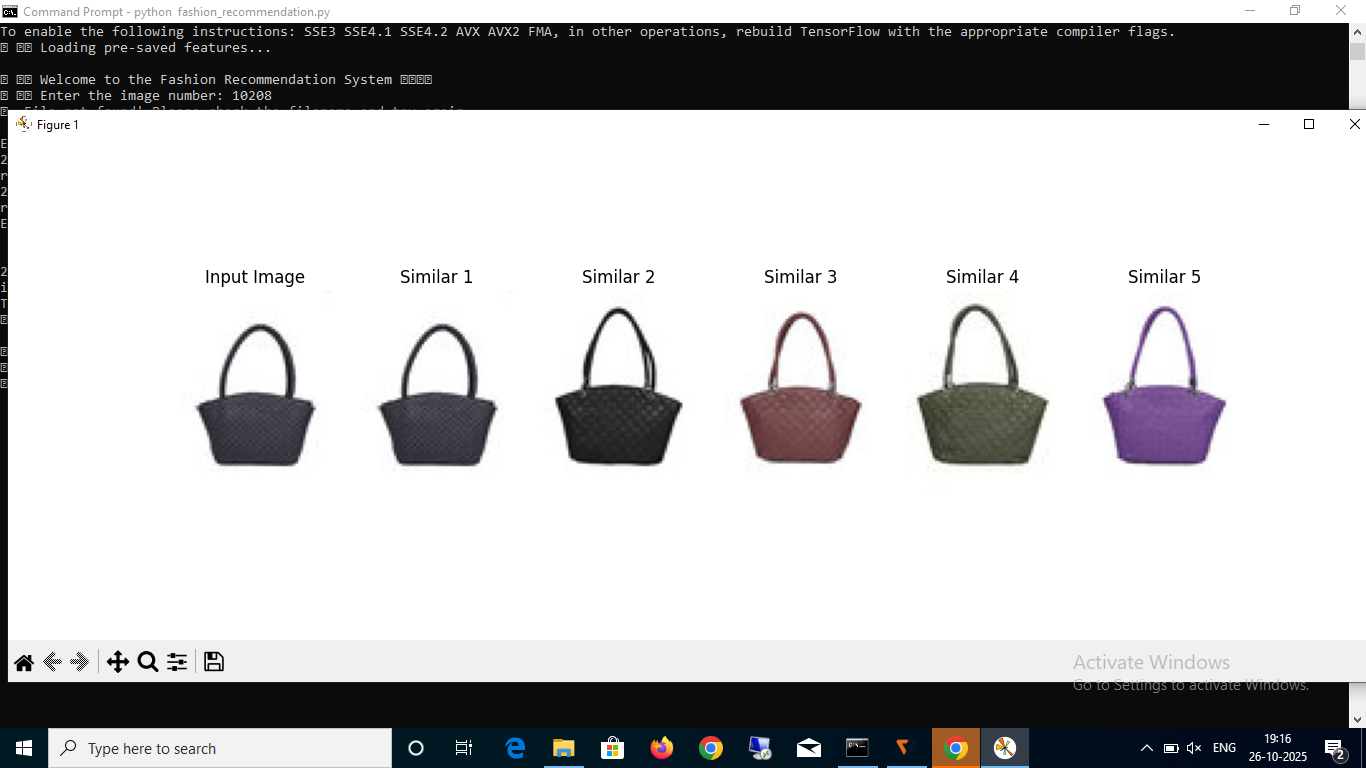
The system is tested with various image inputs from the dataset to verify accuracy in finding similar items. Functional testing ensures that all images are correctly processed, and similarity ranking is accurate.

## CHAPTER – 7 MODULES

* 1. Dataset Loading Module – Reads and prepares image files from the dataset folder.
* 2. Feature Extraction Module – Uses MobileNetV2 for extracting deep visual features.
* 3. Similarity Calculation Module – Calculates cosine similarity between feature vectors.
* 4. Visualization Module – Displays the input image and top similar items.
* 5. User Interaction Module – Accepts user input and triggers the recommendation process.

## CHAPTER – 8 OUTPUT DESIGN

The output displays the input image on the left side and the top five visually similar images on the right. This helps users easily find items with matching designs, colors, or styles.



## CHAPTER – 9 PROJECT CODE

Main Python Program (fashion\_recommendation.py):

import os, cv2  
import numpy as np  
import matplotlib.pyplot as plt  
from tqdm import tqdm  
from sklearn.metrics.pairwise import cosine\_similarity  
from tensorflow.keras.applications import MobileNetV2  
from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input  
from tensorflow.keras.preprocessing import image  
  
DATASET\_PATH = r"data"  
model = MobileNetV2(weights="imagenet", include\_top=False, pooling='avg')  
  
features\_file = "fashion\_features.npy"  
filenames\_file = "fashion\_filenames.npy"  
  
if not os.path.exists(features\_file):  
 print("Extracting features... please wait")  
 features = []  
 filenames = []  
 for file in tqdm(os.listdir(DATASET\_PATH)):  
 if file.lower().endswith(('.jpg', '.png', '.jpeg')):  
 img\_path = os.path.join(DATASET\_PATH, file)  
 img = image.load\_img(img\_path, target\_size=(224, 224))  
 img\_array = image.img\_to\_array(img)  
 img\_array = np.expand\_dims(img\_array, axis=0)  
 img\_array = preprocess\_input(img\_array)  
 feature = model.predict(img\_array, verbose=0)  
 features.append(feature.flatten())  
 filenames.append(img\_path)  
 features = np.array(features)  
 np.save(features\_file, features)  
 np.save(filenames\_file, filenames)  
else:  
 features = np.load(features\_file)  
 filenames = np.load(filenames\_file, allow\_pickle=True)  
  
def show\_similar\_images(img\_path, top\_n=5):  
 img = image.load\_img(img\_path, target\_size=(224, 224))  
 img\_array = image.img\_to\_array(img)  
 img\_array = np.expand\_dims(img\_array, axis=0)  
 img\_array = preprocess\_input(img\_array)  
 feature = model.predict(img\_array, verbose=0).flatten()  
 similarities = cosine\_similarity([feature], features)[0]  
 indices = np.argsort(similarities)[::-1][:top\_n]  
 plt.figure(figsize=(15, 5))  
 plt.subplot(1, top\_n + 1, 1)  
 plt.imshow(image.load\_img(img\_path))  
 plt.title("Input Image")  
 plt.axis('off')  
 for i, idx in enumerate(indices):  
 plt.subplot(1, top\_n + 1, i + 2)  
 plt.imshow(image.load\_img(filenames[idx]))  
 plt.title(f"Similar {i+1}")  
 plt.axis('off')  
 plt.show()  
  
user\_input = input("Enter the image number: ").strip()  
test\_img = os.path.join(DATASET\_PATH, user\_input)  
if os.path.exists(test\_img):  
 show\_similar\_images(test\_img, top\_n=5)  
else:  
 print("File not found!")

## CHAPTER – 10 CONCLUSION

The Fashion Recommendation System effectively demonstrates how deep learning and image processing can be combined to provide smart recommendations. Using MobileNetV2, the system extracts features efficiently and compares them to find similar products. This project serves as a foundation for AI-powered retail applications and online shopping experiences.

## CHAPTER – 11 FUTURE ENHANCEMENT

* Integrate a web interface using Flask or Streamlit.
* Allow filtering by color, texture, and category.
* Add database support for storing product metadata.
* Integrate deep multimodal models like CLIP for advanced similarity detection.
* Deploy the system as a mobile application.