**FASHION RECOMMENDER SYSTEM**

**Submitted for**

**Statistical Machine Learning CSET211**

Submitted by:

**(E23CSEU1723) ISHIKA**

Submitted to:

**DR. NITIN ARVIND SHELKE**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr No. | Content | Page No. |
| 1 | Abstract | 1 |
| 2 | Introduction | 1 |
| 3 | **Related Work** | 1 |
| 4 | **Methodology** | 2 |
| 5 | **Hardware/Software Required** | 2 |
| 6 | **Experimental Result** | 2 |
| 7 | **Conclusions** | 3 |
| 8 | **GitHub Link of Your Complete Project** | 3 |

**1. Abstract**

The **Fashion Recommender System** leverages a deep learning-based content retrieval mechanism to recommend fashion products similar to a user-provided image. Using a pre-trained ResNet50 model, the system extracts image features, normalizes them, and finds the closest matches in a feature vector database using k-Nearest Neighbors (k-NN). The recommender system demonstrates the application of transfer learning and machine learning in personalized fashion retail.

**2. Introduction**

Recommender systems are crucial in modern e-commerce platforms, helping users discover products that align with their preferences. This project focuses on building a **content-based fashion recommender system** using **deep learning**. The system allows users to upload an image, and it retrieves visually similar items from a predefined dataset. The objective is to enhance user experience by providing personalized and intuitive fashion recommendations.

**3. Related Work**

Several works have utilized deep learning and computer vision for content-based image retrieval (CBIR). Transfer learning with pre-trained models like ResNet has been widely applied for feature extraction. Notable advancements include:

* Amazon's "Similar Items" feature for e-commerce recommendations.
* Research on **image embeddings** and similarity search using FAISS or Annoy for scalable solutions.

Our project adapts and implements these principles specifically for fashion recommendations.

**4. Methodology**

**Dataset Preparation**

* The dataset consists of fashion images stored locally in an images folder.
* Features were extracted using the **ResNet50** model and saved as embeddings.pkl.
* Corresponding filenames were saved in filenames.pkl.

**Feature Extraction**

* Pre-trained ResNet50 was used as the backbone for feature extraction, without its top classification layer.
* GlobalMaxPooling2D was added to obtain flattened feature vectors.
* Features were normalized to ensure scale invariance during similarity search.

**Similarity Search**

* k-Nearest Neighbors (k-NN) was used with the **Euclidean distance** metric to identify the top 5 most similar images.
* The brute algorithm was applied for simplicity.

**User Interaction**

* A **Streamlit-based UI** was created for real-time interaction.
* Users can upload an image, and the system displays the top 5 recommendations visually using OpenCV or Streamlit's image display.

**5. Hardware/Software Required**

**Hardware Requirements**

* Processor: Intel i5
* RAM: 8 GB
* GPU: NVIDIA GPU (for faster inference)

**Software Requirements**

* Python: 3.8
* Libraries: TensorFlow, NumPy, OpenCV, Streamlit, scikit-learn
* Operating System: Windows

**6. Experimental Results**

The system was tested with a dataset of 40000+ fashion images.

* The accuracy of the recommendations was evaluated qualitatively based on visual similarity.
* Average response time for recommendation generation was ~2 seconds on a CPU.
* Recommendations were consistent and aligned with user expectations.

**7. Conclusions**

The Fashion Recommender System effectively demonstrates the potential of deep learning and content-based retrieval for personalized recommendations. By using pre-trained models, we achieved robust feature extraction without extensive computational resources.

**8. Future Scope**

* **Dataset Expansion**: Increase the dataset size and diversity for improved recommendations.
* **Scalability**: Integrate approximate nearest neighbor (ANN) methods like FAISS for faster retrieval on large datasets.
* **Metadata Integration**: Combine visual features with metadata (e.g., color, brand) for hybrid recommendations.
* **Cross-Platform Deployment**: Deploy the system as a web or mobile application for broader accessibility.
* **Real-Time Updates**: Enable dynamic updates to the feature database to accommodate new items.

**9. GitHub Link of Your Complete Project**

[GitHub Repository](https://github.com/your-repo-link) : https://github.com/ishikabeniwal/ishika