PROJECT ABSTRACT

This project presents a sophisticated Fashion Recommender System, harnessing the power of ResNet-50 architecture, TensorFlow, and Deep Learning to revolutionize personalized fashion recommendations. ResNet-50, a renowned deep convolutional neural network, is employed for feature extraction, enabling the system to capture intricate details in fashion images. TensorFlow, a powerful open-source machine learning framework, facilitates seamless model development, training, and deployment. The synergy of these technologies results in a robust and efficient recommendation system.

Our Deep Learning model is trained on vast datasets, learning complex patterns and styles inherent in fashion images. Through an iterative process, the model refines its understanding of user preferences, ensuring a personalized and accurate recommendation experience. The system goes beyond traditional recommender systems by incorporating visual cues, enabling users to discover fashion items that align with their unique tastes.

The implementation of ResNet-50 ensures superior image recognition capabilities, allowing the model to discern subtle nuances in style, color, and design. TensorFlow's scalability and efficiency enable real-time processing, making the recommender system suitable for dynamic fashion trends and user interactions. The proposed system not only enhances user satisfaction but also contributes to a more engaging and tailored fashion discovery experience. This project represents a groundbreaking approach to fashion recommendation, leveraging state-of-the-art technologies to create a cutting-edge solution at the intersection of deep learning and fashion.

I. INTRODUCTION

In the ever-evolving landscape of technology and fashion, our project endeavours to bridge the gap between these two realms through the development of an innovative Fashion Recommender System. This project explores the synergy of ResNet-50 architecture, TensorFlow, and Deep Learning techniques to revolutionize the way individuals discover and engage with fashion.

ResNet-50, a powerful convolutional neural network renowned for its deep learning capabilities, serves as the cornerstone of our project. By leveraging its ability to extract intricate features from images, we aim to create a robust foundation for our Fashion Recommender System. This approach allows the model to discern subtle details, patterns, and styles within fashion images, providing a comprehensive understanding of the visual elements that define individual preferences.

The integration of TensorFlow, a versatile and scalable machine learning framework, facilitates the seamless development and deployment of our recommendation system. TensorFlow's efficiency in handling large datasets and real-time processing aligns seamlessly with the dynamic nature of fashion trends and user interactions.

Deep Learning plays a pivotal role in our project by enabling the model to continuously learn and adapt to evolving fashion preferences. Through extensive training on diverse datasets, our system evolves iteratively, ensuring that it not only captures prevalent trends but also tailors recommendations to each user's unique style profile.

This project represents a significant exploration into the intersection of technology and fashion, aiming to enhance user experiences and redefine how individuals engage with their personal style. The fusion of ResNet-50, TensorFlow, and Deep Learning methodologies promises to deliver a cutting-edge Fashion Recommender System that transcends conventional approaches, paving the way for a more personalized and engaging fashion discovery journey.

OBJECTIVE

The central objective of our project is to design and implement an innovative Fashion Recommender System that combines the strengths of ResNet-50 architecture, K nearest Neighbors (KNN), TensorFlow, and Deep Learning methodologies. This project aims to redefine the fashion recommendation paradigm by offering a comprehensive solution that considers both visual features and user preferences.

The integration of ResNet-50, a powerful deep learning model, serves as the foundation for robust feature extraction from fashion images. This ensures a detailed understanding of clothing styles, patterns, and nuances, contributing to the accuracy of personalized recommendations. The inclusion of KNN complements this approach by incorporating collaborative filtering, enabling the system to leverage the preferences of similar users to enhance recommendation precision.

TensorFlow, a versatile machine learning framework, plays a vital role in the seamless development and deployment of the recommendation system. Its scalability and efficiency are crucial for handling large datasets and real-time processing, ensuring the system remains adaptive to the dynamic nature of fashion trends.

Deep Learning methodologies further enhance the system's capabilities by facilitating continuous learning and adaptation. Through extensive training on diverse datasets, the system evolves iteratively, staying attuned to evolving fashion preferences and emerging trends.

In essence, our project aims to pioneer a Fashion Recommender System that transcends traditional approaches by integrating ResNet-50, K nearest Neighbors, TensorFlow, and Deep Learning. By amalgamating visual and collaborative filtering techniques, we strive to provide users with a sophisticated and tailored fashion discovery experience, revolutionizing how individuals engage with their personal style preferences, aim to provide users with an unparalleled fashion discovery experience tailored to their unique styles and preferences.

PROBLEM STATEMENT

In the realm of fashion e-commerce, current recommender systems often fall short in providing truly personalized and nuanced recommendations, creating a gap between user expectations and system capabilities. The existing approaches predominantly rely on singular methods, lacking a holistic integration of advanced technologies. Therefore, the problem at hand is to address these limitations by developing a comprehensive Fashion Recommender System that leverages the synergies of ResNet-50, K nearest Neighbors (KNN), TensorFlow, and Deep Learning.

Existing systems struggle to capture the intricacies of user preferences, as they often overlook the visual subtleties inherent in fashion choices. Conventional recommendation algorithms lack the depth required to analyze diverse clothing styles, patterns, and nuanced visual features. Additionally, collaborative filtering, a key aspect in enhancing recommendations based on user similarity, is often underutilized.

The challenge lies in bridging these gaps and creating a unified system that not only extracts rich features from fashion images using ResNet-50 but also incorporates collaborative filtering through KNN. Furthermore, seamlessly integrating these technologies using TensorFlow poses a challenge, requiring careful consideration of scalability, real-time processing, and user interaction dynamics. Deep Learning methodologies must be adeptly employed to enable continuous learning, ensuring the system remains agile in adapting to evolving fashion trends and user preferences.

Thus, the problem statement revolves around the need for a sophisticated Fashion Recommender System that amalgamates ResNet-50, KNN, TensorFlow, and Deep Learning, addressing the shortcomings of existing systems and providing users with a truly personalized and dynamic fashion discovery experience.

II. Proposed System

Our proposed Fashion Recommender System seeks to redefine the landscape of personalized fashion discovery by integrating state-of-the-art technologies, including ResNet-50, K nearest Neighbors (KNN), TensorFlow, and Deep Learning methodologies. At the core of our system lies ResNet-50, a robust deep learning architecture renowned for its ability to extract intricate features from fashion images. This enables a granular understanding of clothing styles, textures, and visual elements.

The incorporation of KNN introduces collaborative filtering, enhancing the model by leveraging user similarities. By analyzing patterns in user preferences, the system intelligently refines its suggestions, creating a more tailored and accurate experience. TensorFlow, a versatile machine learning framework, serves as the backbone, ensuring seamless model development, scalability, and real-time processing capabilities.

Deep Learning methodologies play a pivotal role, enabling continuous learning and adaptation to evolving fashion landscapes. Through extensive training on diverse datasets, the system evolves iteratively, staying abreast of emerging trends and refining its understanding of user preferences.

Our proposed system envisions a dynamic synergy of these technologies, offering users a sophisticated and personalized fashion discovery journey. By combining the visual richness of ResNet-50, the collaborative filtering capabilities of KNN, the versatility of TensorFlow, and the adaptive learning of Deep Learning, we aim to deliver a groundbreaking Fashion Recommender System that not only meets but exceeds user expectations, redefining the way individuals engage with and explore their unique fashion preferences.

III. Software and Hardware Requirements

The implementation of our Fashion Recommender System, integrating ResNet-50, K nearest Neighbors (KNN), TensorFlow, Deep Learning, Neural Networks, pandas, and Streamlit, necessitates a robust infrastructure and a suite of specialized software tools.

Hardware Requirements:

To support the computational demands of deep learning and neural network operations, a high-performance GPU (Graphics Processing Unit) is recommended. A system with a multi-core processor, sufficient RAM (Random Access Memory), and ample storage space is essential for efficient model training and real-time processing. Additionally, reliable internet connectivity is crucial for accessing and updating datasets.

Software Requirements:

Python Environment: The project relies on Python as the primary programming language. A virtual environment, managed using tools like Anaconda or virtualenv, ensures a controlled and isolated development environment.

Deep Learning Frameworks: TensorFlow is a critical component for model development, training, and deployment. The installation of TensorFlow-GPU enhances performance, leveraging the capabilities of the GPU.

Neural Network Libraries: Integration of neural networks requires libraries such as Keras, a high-level neural networks API that seamlessly interfaces with TensorFlow.

Data Processing and Analysis: Pandas, a data manipulation and analysis library, is employed for handling datasets, preprocessing, and extracting relevant features for training.

Streamlit: For creating an interactive and user-friendly interface, Streamlit is utilized to develop the front-end of the application, enabling seamless visualization and interaction with the recommender system.

ResNet-50 Pre-trained Model: The implementation involves leveraging a pre-trained ResNet-50 model, accessible through frameworks like Keras applications, saving time and resources during the development phase.

The amalgamation of these software and hardware components forms the technological backbone for our Fashion Recommender System, ensuring a robust and efficient platform for the integration of cutting-edge technologies in the domain of fashion recommendation.

IV. SCREENSHOTS

```
import tensorflow
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50,preprocess_input
```

```
WARNING:tensorflow:From C:\Users\ds448\anaconda3\Lib\site-packages\keras\src\
cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_c
```

```
model=ResNet50(weights='imagenet',include_top=False,input_shape=(224,224,3))
```

```
import pickle
pickle.dump(filenames,open('filenames.pkl','wb'))
pickle.dump(fe_list,open('embeddings.pkl','wb'))
```

```
from numpy.linalg import norm
img=image.load_img('sample/tshirt1.jpeg',target_size=(224,224))
img_arr=image.img_to_array(img)
exp_img_arr=np.expand_dims(img_arr,axis=0)
prep_img=preprocess_input(exp_img_arr)
final=model.predict(prep_img).flatten()
norm_res=final/norm(final)
```

```python
from sklearn.neighbors import NearestNeighbors
neighbors=NearestNeighbors(n_neighbors=6,algorithm='brute',metric='euclidean')
neighbors.fit(fe_list)
```

```
NearestNeighbors(algorithm='brute', metric='euclidean', n_neighbors=6)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```python
distances,indices=neighbors.kneighbors([norm_res])
```

```python
for file in indices[0]:
    print(filenames[file])
```

```python
import streamlit as st
import os
from PIL import Image
import numpy as np
import pickle
import tensorflow
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50,preprocess_input
from sklearn.neighbors import NearestNeighbors
from numpy.linalg import norm


feature_list = np.array(pickle.load(open('embeddings.pkl','rb')))
filenames = pickle.load(open('filenames.pkl','rb'))

model = ResNet50(weights='imagenet',include_top=False,input_shape=(224,224,3))
model.trainable = False

model = tensorflow.keras.Sequential([
    model,
    GlobalMaxPooling2D()
])
```

```
uploaded_file = st.file_uploader("Choose an image")
if uploaded_file is not None:
    if save_uploaded_file(uploaded_file):
        # display the file
        display_image = Image.open(uploaded_file)
        st.image(display_image)
        # feature extract
        features = feature_extraction(os.path.join("uploads",uploaded_file.name),model)
        #st.text(features)
        # recommendention
        indices = recommend(features,feature_list)
        # show
        col1,col2,col3,col4,col5 = st.columns(5)

        with col1:
            st.image(filenames[indices[0][0]])
        with col2:
            st.image(filenames[indices[0][1]])
        with col3:
            st.image(filenames[indices[0][2]])
        with col4:
            st.image(filenames[indices[0][3]])
        with col5:
            st.image(filenames[indices[0][4]])
    else:
        st.header("Some error occured in file upload")
```

```
def recommend(features,feature_list):
    neighbors = NearestNeighbors(n_neighbors=5, algorithm='brute', metric='euclidean')
    neighbors.fit(feature_list)
    distances, indices = neighbors.kneighbors([features])

    return indices
```

V. LIMITATIONS

While our Fashion Recommender System represents a significant advancement in personalized fashion discovery, it is crucial to acknowledge certain limitations inherent in the current implementation.

Dependency on Data Quality: The effectiveness of our recommender system heavily relies on the quality and diversity of the training data. In cases where the dataset is limited in scope or biased towards specific fashion trends, the system may struggle to provide well-rounded recommendations, potentially limiting its applicability across diverse user preferences.

Computational Intensity: The integration of deep learning technologies, including ResNet-50 and neural networks, demands substantial computational resources. While high-performance GPUs optimize model training, the system's efficiency may be constrained by hardware limitations, potentially leading to longer processing times and increased computational costs.

Interpretability and Explainability: The complex nature of deep learning models, including ResNet-50 and neural networks, may result in limited interpretability and explainability. Users may find it challenging to understand the rationale behind specific recommendations, potentially affecting their trust in the system.

Limited Feature Engineering: While the integration of pandas facilitates data preprocessing, the system's reliance on automated feature extraction from deep learning models may limit the incorporation of explicit user-driven features. This could potentially impact the system's ability to capture nuanced aspects of individual fashion preferences.

User Interface Constraints: While Streamlit provides an intuitive and interactive user interface, the system's user experience is contingent on the design and responsiveness of the interface. Limitations in the interface design may affect user engagement and the overall effectiveness of the recommendation system.

VI. CONCLUSION

In the culmination of our endeavor to create a state-of-the-art Fashion Recommender System, the integration of ResNet-50, K nearest Neighbors (KNN), TensorFlow, Deep Learning, Neural Networks, pandas, and Streamlit has resulted in a comprehensive solution that pushes the boundaries of personalized fashion discovery.

The amalgamation of deep learning and neural network technologies, facilitated by TensorFlow and Keras, empowers the system to discern intricate visual details through ResNet-50's feature extraction capabilities. The inclusion of KNN for collaborative filtering enhances recommendation accuracy by considering user similarities, providing a more nuanced and personalized experience.

Pandas, a versatile data processing library, enables efficient handling and manipulation of datasets, contributing to the system's robustness. Streamlit, with its intuitive interface design, elevates user engagement, making the Fashion Recommender System accessible and user-friendly.

However, our project is not without its limitations. Challenges related to data quality, computational intensity, and interpretability must be acknowledged for future enhancements. Despite these constraints, our Fashion Recommender System represents a significant leap forward in leveraging cutting-edge technologies for fashion recommendation.

In conclusion, the successful integration of ResNet-50, KNN, TensorFlow, Deep Learning, Neural Networks, pandas, and Streamlit has resulted in a sophisticated and user-centric platform. This project lays the foundation for further advancements in the fusion of technology and fashion, promising a more tailored and dynamic fashion discovery experience for users in the ever-evolving landscape of e-commerce.

REFERENCES

[1] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

[2] https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/

[3] https://keras.io/api/

[4] https://thecleverprogrammer.com/2020/08/16/fashion-recommendation-system/

[5] https://www.scaler.com/topics/course/deep-learning-free-course/