FASHION RECOMMENDATION USING DEEP LEARNING TECHNIQUES

Sampath Kumar Garminla, Gowthami Middela, Ajay Kumar Reddy Vemapati

Abstract:

Recommendation systems are methods for forecasting a person's assessment of a product or social organization. Books, movies, dining establishments, and other items with different tastes can be added. To forecast these preferences, two methods are applied. The first method is content-based and makes use of item features, whereas the second method is collaborative and evaluates the user's selections based on past behavior. A fashion recommendation system based on the style classification of the input apparel images is presented in this thesis. In this article, the emphasis is on images of models wearing clothing on people. We used webpage scraping to create our databases.

Keywords: Python, TensorFlow, Keras, CNN, Web scraping, ResNet-50.

Introduction

By combining the ResNet-50 architecture with Annoy, an enhanced K-Nearest Neighbors algorithm, the fashion recommendation system leverages the potential of transfer learning to provide customized suggestions based on user input. By doing feature extraction on a large dataset of more than 45,000 photographs using transfer learning with ResNet-50, I was able to evaluate picture data efficiently. In order to provide individualized fashion recommendations, I created a similarity search technique using K-Nearest Neighbors to identify the top 5 closest matches to a user's input. The system is easy to use and intuitive, making it possible to analyze image data accurately and effectively.

By showcasing the adaptability and effectiveness of similarity search, convolutional neural networks (CNNs), and transfer learning, this recommendation system paves the way for bigger and more intricate systems.

An entire suite of web apps that forecasts user reactions to options is called a recommendation program. The recommendation system has been a popular topic for a while. Recommendation techniques, or recommendation programs, are essentially straightforward algorithms that filter important items from a vast array of information sources in an effort to provide the user with the most accurate and pertinent information possible. By looking at customer preferences and generating results that are pertinent to their requirements and interests, recommendation engines identify patterns in data collections. A recommendation engine is used in real-time instances, like Amazon, to suggest products to users that they might find interesting. Content-based filtering strategies are determined by the item description and the user preference profile. These methods work best when an object has known information (name, location, description, etc.), but not the user.

Content-based recommendation learns a user's preferences based on the attributes of the item and approaches suggestion as a specific user problem. The algorithms try to suggest items based on what has been judged to be comparable to the user's preferences in the past or present. A user's login computer did not create this temporary profile. In example, a number of potential items are strongly

recommended after being compared to items that have received user ratings. This approach is founded on research in data filtering and information retrieval.

The objective of the research is to develop a model that can propose clothing based only on its appearance. Before endorsing an image, the model takes a look at it to see if it features any fashion items. This project's main objective is to:

• Develop a method for recommending clothes that provides guidance on what to buy.

• To identify the sort of fashion shown in an input photograph.

• If the fashion photograph that was supplied is legitimate, a matching outfit will be suggested.

• Obtaining goods based on comparable search queries from different e-commerce sites.

Related work

There have been some earlier studies on creating recommendation systems. Intelligent Clothes Suggestion System Utilizing Deep Learning [1] We create two inception-based convolutional neural networks for the prediction portion and one feed forward neural network for the recommender in order to recommend a cloth. In this investigation, we were able to predict colors with 98% accuracy, gender and fabric patterns with 86% accuracy, and wardrobe recommendations with 75% accuracy.

Decomposition of Style Features in a Deep Fashion Recommendation System [2] However, because style and category information are inconsistent, the clothes vector frequently suggests mismatched items. We suggest a style feature extraction (SFE) layer as a solution to this issue, which efficiently separates the garments vector into style and category. We extract and remove the category information from the garments vector in order to acquire more precise style information, based on the features the category information has slight fluctuations in the same class while being distinct from other classes.

Recommendation Engine: Proposed Methodology

In this research, we provide a model that combines the Nearest Neighbor backed recommender with Convolutional Neural Network. According to the figure After training the neural networks, an inventory is chosen to generate suggestions, and a database is made for the inventory items. Based on the supplied image, the nearest neighbor's algorithm locates the most pertinent products and generates suggestions.

Image pre-processing: Image processing is the act of manipulating digital images using computer algorithms. Refining the image data by eliminating distorted noise and improving image pixels is the primary objective of image processing. A two-dimensional array of numbers between 0 and 255 is all that an image is. The mathematical function f(x,y), where x and y are the two horizontal and vertical coordinates, defines it. The pixel value of an image is given by the value of f(x,y) at any given place.

Pre-processing the image involves the following steps:

The user-provided image is read as input and saved in a temporary folder on the server.

• Resize image: The saved image is shrunk to match the input scale (224 x 224) that the model is trained with.

• Segmentation: To improve feature extraction, the stored image is transformed from RGB to BGV at this stage.

• Flatten: At this point, the image's 2D matrix is transformed into a vector after pre-processing the saved file.

Recommendation Engine: A recommendation engine suggests suitable products to users by employing various algorithms to filter the information. It starts by analyzing a customer's historical behavior and makes product recommendations that the customers are likely to purchase. The recommendation engine functions as follows:

• Gathering Information: The recommendation engine is created first by gathering data. There are two types of data: implicit and explicit. User input, such as product evaluations and comments, might be explicit data. Additionally, the implicit data would be the order history/return history, cart events, page views, click through, and search log. Every visitor to the website will have a dataset built for them.

• Analyzing the Data: Real-time system analysis is used to filter the data. As data is created, it can be processed by real-time systems. Typically, this system makes use of technologies for processing and analyzing event streams. Making recommendations on the spot is a must.

• Filtering the Data: The following action is to filter the data to obtain the information required to give the user recommendations. In this project, a content-based filtering strategy is employed. Content-based filtering makes recommendations for additional things that are comparable to the user's preferences based on explicit feedback or past actions, using meta data or item characteristics.

• Web scraping: This quick and effective method of automating data extraction is known as web scraping.It makes use of crawlers, or robots, which automatically scan particular web pages and retrieve the necessary data. We use a piece of code known as a "web scraper" to request a specific product page on an e-commerce website in order to gather product data on a wide scale. The website responds by displaying the webpage that was requested. The scraper will parse the page's HTML code and collect pertinent data as soon as it is received. The tool then transforms the data into the required format after the data extraction process is finished.

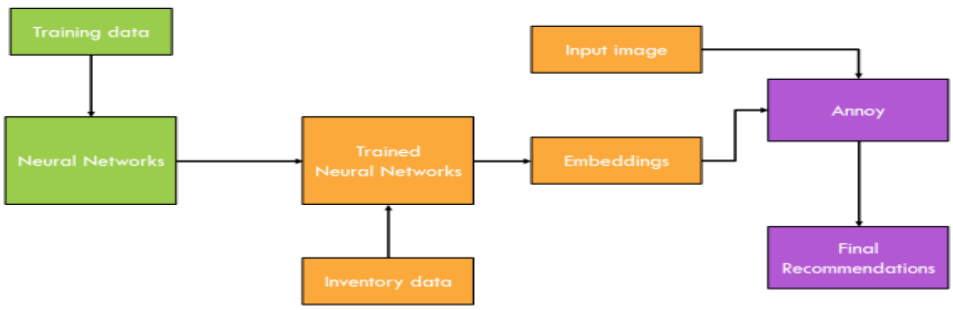
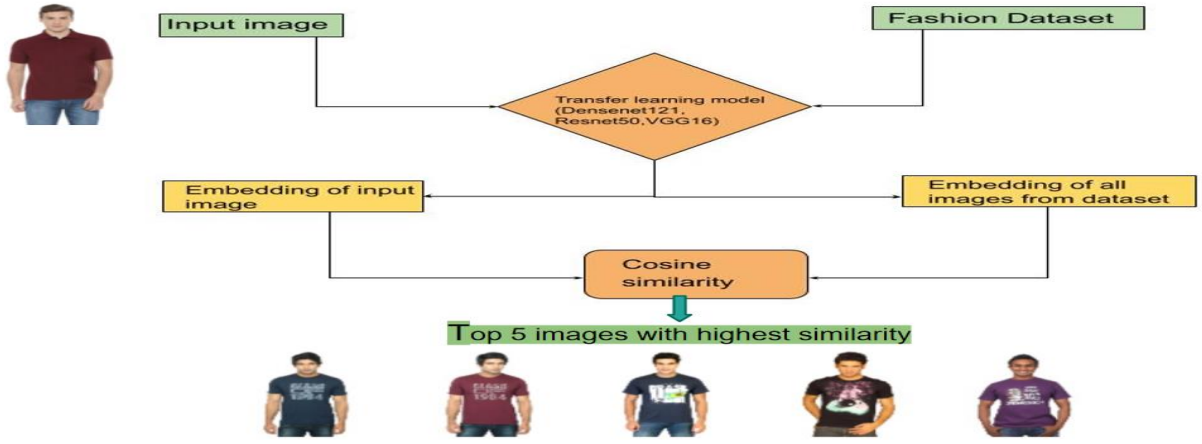


Figure 1. Block diagram of proposed system

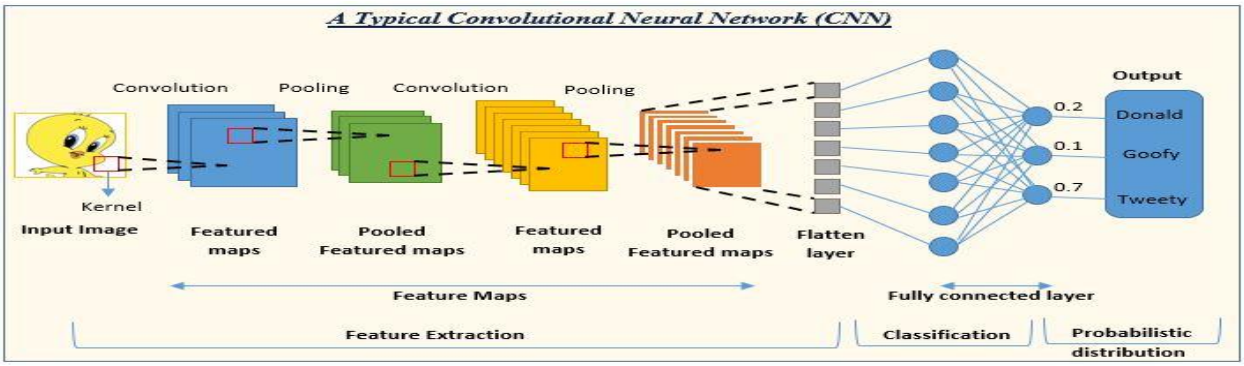
Flow-Chart:

In our suggested approach, Sklearn Nearest Neighbors is employed to generate suggestions. This aids in our search for the input image's closest neighbors. In this study, the Cosine Similarity measure was utilized. The database's top five suggestions are selected, and their images are displayed.



Convolutional Neural Networks:

A particular kind of neural network designed for processing visual data, like images and videos, is called a convolutional neural network. CNNs, however, are also effective with non-image data (especially in text categorization and natural language processing). Because it functions on the same general principle of propagation in the forward and backward directions as a typical neural network (multilayer perceptron), its concept is similar to that of a neural network. ResNet50 transfer learning is used to train the neural networks following the pre-processing of the data. In the last stages, additional layers are added to replace the ResNet50 design, and weights are adjusted to optimize the network model for the current circumstances. The figure shows the architecture of ResNet50.



Some earlier works have been connected to the advancement of recommendation systems.

The Deep Learning Apparel Suggestion System [1] To recommend a fabric, we build one feed-forward neural network as a recommender and two inceptions-based convolutional neural networks as predictive components. In this investigation, we were able to predict color with 98% accuracy, gender and textile pattern with 86% accuracy, and apparel suggestions with 75% accuracy.

Decomposition of Style Features in a Deep Fashion Recommendation System [2] However, the outfits vector often suggests mismatched clothing because of the variable understanding of style and category. We propose a style feature extraction (SFE) layer to tackle this problem, which efficiently breaks down the clothes vector into style and category. Considering the features of the category data, To get more precise style information, we extract and remove the category information from the garments vector.

IMPLEMENTATION

Image Preprocessing: Improving the quality of the image is the primary goal of image processing, which also involves feature extraction and categorization. The most popular fields in which it is employed include computer vision, imaging in medicine, meteorology, astronomy, remote sensing, and related fields. Instruments for image processing:

• OpenCV: This program allows us to read and write images, record and store movies, and manipulate (filter, transform) images.

• Numpy: A library of multidimensional array objects, Numpy (an acronym for Numerical Python)

```
def feature_extraction(img_path,model):
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    expanded_img_array = np.expand_dims(img_array, axis=0)
    preprocessed_img = preprocess_input(expanded_img_array)
    result = model.predict(preprocessed_img).flatten()
    normalized_result = result / norm(result)

    return normalized_result


def recommend(features,feature_list):
    neighbors = NearestNeighbors(n_neighbors=6, algorithm='brute', metric='euclidean')
    neighbors.fit(feature_list)

    distances, indices = neighbors.kneighbors([features])

    return indices

# steps
# file upload -> save
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(uploadn_path,uploaded_file.name),model)
        # 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:
```

Web Scraping

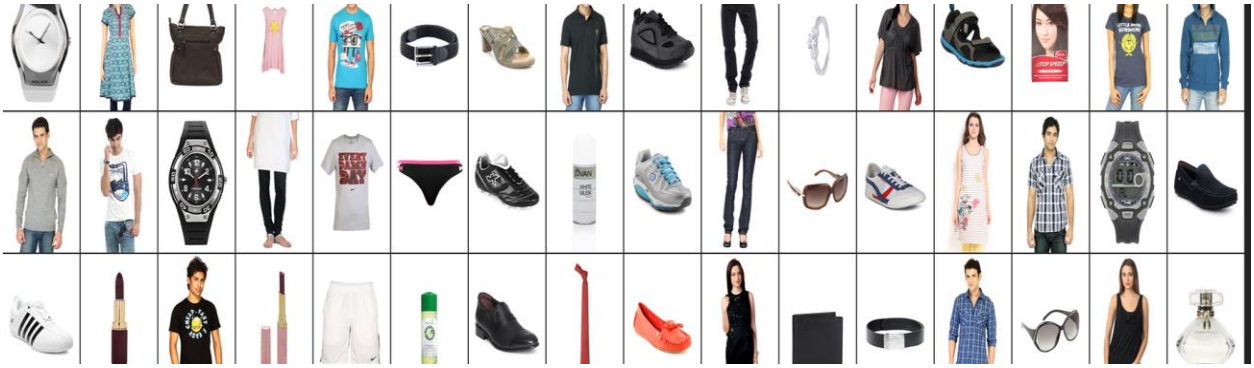
Massive volumes of data are extracted and processed from the web via web scraping. The capacity to quickly obtain vast volumes of data from the internet is made possible by the ability to scrape data from the website. One crucial method for extracting data into a format that is useful for data analysis is web scraping with Python.

```
from setuptools import setup

with open("README.md", "r", encoding="utf-8") as f:
    long_description = f.read()

setup(
    name="src",
    version="0.0.1",
    author="Bappy Ahmed",
    description="A small package for Fasion Recommendation system Deep Learning Project",
    long_description=long_description,
    long_description_content_type="text/markdown",
    url="https://github.com/entbappy/",
    author_email="entbappy73@gmail.com",
    packages=["src"],
    python_requires=">=3.7",
    install_requires=[
        'tensorflow==2.3.1',
        'PyYAML',
        'scikit-learn',
        'tqdm',
        'streamlit',
        'opencv-python',
        'bing-image-downloader'
    ]
)
```

```
import os

def execute_system():
    bash1 = 'python src/01_generate_embedding.py'
    os.system(bash1)

    print('Executed successfully!! Now run app.py')

if __name__ == '__main__':
    execute_system()
```

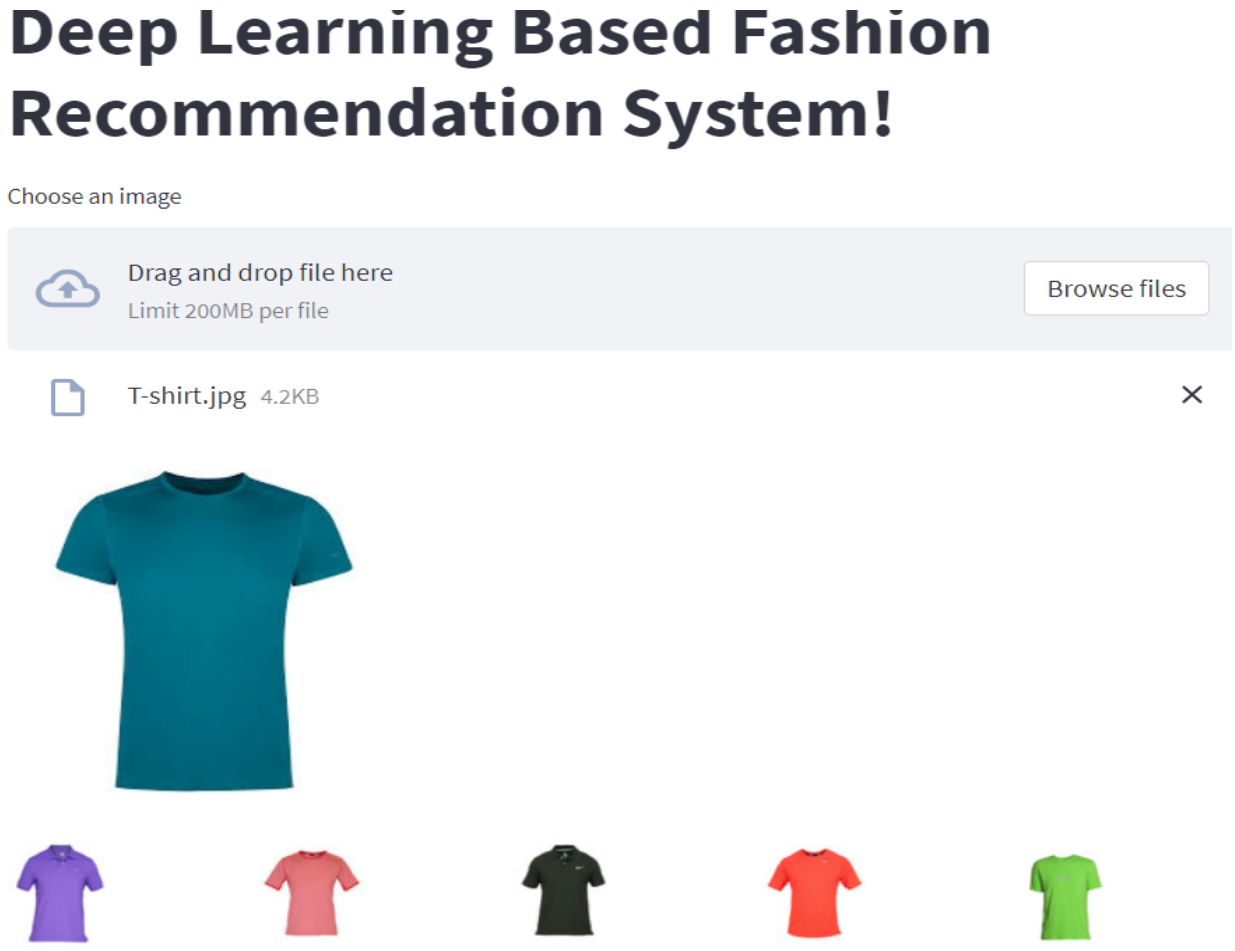
The Figure shows a sample set of inventory data:

The pictures are from the Product Images Dataset on Kaggle Fashion. After the inventory has been classified and embedded using neural networks, recommendations are generated based on the output.



Result

Users of this web application can locate products by using a picture. It offers a user-friendly interface that makes it easier for users to locate products that visually appeal to them. All it takes is clicking on an image of the product to upload it to the web application. Products that the model thought were similar to the uploaded image will be shown to the user. If they so choose, consumers can purchase the product from the corresponding e-commerce websites.



Conclusion:

Engines for product suggestions are the most effective means of providing consumers with a better overall experience. Product recommendations engines can assist in providing clients with relevant products that they need or require by utilizing machine learning, manual curation, and targeted algorithms.It enables marketers to instantly offer consumers recommendations for products that are relevant to them. Product recommendations are a component of e-commerce customization strategies that improve the consumer experience by dynamically populating products onto websites, applications, contact centers, or emails.

In this study, we have introduced a revolutionary framework for data-driven, visually-related, and straightforward recommendation algorithms for fashion product picture generation. The suggested method employs a two-phase process. Our suggested method first uses a CNN classifier to extract the image's features. For example, it lets users upload any random fashion image from any e-commerce website, and then it generates images that are similar to the uploaded image based on the texture and features of the input image. Further research is necessary to provide more accurate recommendations and enhance the whole fashion exploration experience for both direct and indirect consumers.

Reference:

1. Han, Xintong & Wu, Zuxuan & Jiang, Yu-Gang & Davis, Larry. (2017). Learning Fashion Compatibility with Bidirectional LSTMs. 10.1145/3123266.3123394.
2. Liu, Si & Feng, Jiashi & Song, Zheng & Zhang, Tianzhu & Lu, Hanqing & Xu, Changsheng & Yan, Shuicheng. (2012). Hi, magic closet, tell me what to wear!. 619-628. 10.1145/2393347.2393433.
3. Jiang, Shuhui & Fu, Yun. (2017). Fashion Style Generator. 3721-3727. 10.24963/ijcai.2017/520.
4. https://www.researchgate.net/publication/360087895_Fashion_recommendation_system_using_CNN