IMAGE-BASED RECOMMENDATION USING RESNET

Aswathy Babu
Department Of Computer Applications
Amal Jyothi College of Engineering
Kanjirappally, India
aswathybabu2023a@mca.ajce.in

Amal K Jose
Department Of Computer Applications
Amal Jyothi College of Engineering
Kanjirappally, India
amalkjose@amaljyothi.ac.in

***Abstract*— The goal of this project is to provide product recommendations using image search. The recommendations will be provided on the basis of image similarity using the ResNet50 model. The ResNet model is a technique that uses deep learning to recommend items based on their visual similarity. The system allows users to upload an image of a product and then retrieves similar products from a database based on the visual features extracted from the image using the ResNet50 model. The nearest neighbor algorithm is used to find the most similar products to the uploaded image based on the Euclidean distance between the feature vectors. The top five most similar products are returned and displayed to the user.**

Keywords- Image Search, NearestNeighbors, ResNet Model, Recommendations

I. INTRODUCTION

Image-based recommendation systems have lately gained popularity as a result of the expansion of e-commerce platforms and the availability of visual data. The purpose of these systems is to make visually comparable suggestions. ResNet (Residual Network) is one of the most common models for image-based recommendation. ResNet is a deep convolutional neural network (CNN) architecture that is commonly used for image categorization applications.

The fundamental concept underlying image-based recommendations using ResNet is to extract sophisticated representations from images by employing the ResNet model, which are then utilized to produce personalized suggestions. In comparison to traditional recommendation systems that are based on either user-item interactions or content-based filtering, this approach offers several benefits. Image-based recommendations have the potential to present more precise and diverse recommendations by utilizing visual information that may not be identifiable through text-based characteristics. Furthermore, ResNet serves as a highly efficient feature extractor capable of effectively capturing both user preferences and item characteristics. The process under consideration has the capacity to produce intricate visual presentations. By providing a visually appealing interface and facilitating the discovery of novel products through visual exploration, the implementation of image-based recommendations has been shown to enhance the user experience.

II. RESIDUAL NETWORK

The Residual Network, commonly referred to as ResNet, is a neural network architecture characterized by deep layering. Its origination can be credited to Kaiming He et al. In the year 2015. The ResNet50, a derivative of the ResNet architecture that incorporates a network depth of 50 layers, has found significant utility in image search and product recommendation systems. The ResNet50 model is a pre-trained deep learning architecture frequently utilized for image categorization assignments. The utilization of image feature extraction techniques enables the generation of embeddings, which subsequently facilitates the identification of analogous products in search processes.

III. TRANSFER LEARNING

Transfer learning is a deep learning technique in which a previously trained model is used as the foundation for a new model. Transfer learning can save time and computational resources by allowing the new model to use the learned characteristics of the pre-trained model. In computer vision problems, ResNet50 is frequently used as a pre-trained model for transfer learning. The pre-trained model is used as a feature extractor in transfer learning with ResNet50, and the output of the last layer is used as input for a new model. The pre-trained characteristics are then used to train the new model for the new assignment.

IV. LITERATURE REVIEW

A fast-expanding area of research focuses on product suggestion using image search with the goal of enhancing customers' online shopping experiences by offering tailored

DOI: 10.5281/zenodo.7949796
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam

recommendations based on their aesthetic preferences. This method analyses product photos and extracts pertinent information using deep learning models such as convolutional neural networks (CNNs) and the ResNet model. Instead of depending just on information or product descriptions, these attributes are then used to locate comparable products based on visual similarity. E-commerce businesses can boost consumer engagement, boost revenues, and boost customer happiness by using image-search-based product recommendation systems. The use of multi-modal deep learning models for product recommendations and the blending of user behaviour and visual preferences to produce personalised recommendations are only a few current works in this field.

The study described by S. Hussain et al. [1] (2021) focuses on creating an image-based product recommendation system using deep learning techniques. The ResNet50 model is used in the study to construct embeddings and extract features from product photos. The similarity scores between products are then determined using the embeddings, and users are subsequently given recommendations for products based on their search history. The suggested solution performs better than other cutting-edge approaches and achieves great accuracy.

A deep learning-based product recommendation system is proposed by R. Li et al. [2] (2021), utilising a hybrid model that combines ResNet50 with the attention mechanism. A vector representation created by the model after it has extracted picture data is then merged with additional features, such as user history and product attributes, to produce suggestions. The suggested system achieves excellent accuracy and can manage a sizable number of users and items.

In another study, Z. Huang et al. [3] (2021) propose a deep learning-based recommendation system that uses ResNet50 to extract visual features from product images and a graph neural network to model user-item interactions. The proposed model is able to handle sparse data and outperforms other state-of-the-art methods.

The study conducted by M. V. K. S. K. Mahesh et al. [4] (2021) proposes an image-based product recommendation system that uses ResNet50 to extract features from product images and a collaborative filtering algorithm to generate recommendations. The proposed system achieves high accuracy and is able to handle a large number of products and users.

In a study by S. Vijayakumar et al. [5] (2021), they propose a deep learning-based product recommendation system that uses ResNet50 to extract image features and a convolutional neural network to model user-item interactions. The proposed system is able to handle a large number of products and users and achieves high accuracy compared to other state-of-the-art methods.

In another study by G. Zhang et al. [6] (2021), they propose a visual search and recommendation system that uses ResNet50 to extract visual features from product images and a graph-based algorithm to model user-item interactions. The proposed system achieves high accuracy and is able to handle a large number of products and users while providing personalised recommendations based on user preferences.

V. MOTIVATION

First and foremost, e-commerce platforms are fast-growing, and online shopping has assimilated into every aspect of our lives. However, it can be difficult and time-consuming to choose the right item from a large inventory. By streamlining and enhancing the shopping experience, the addition of an image search capability and product recommendations based on uploaded photographs can dramatically enhance customer satisfaction.

The ResNet model can be a useful tool in creating such a system due to its remarkable performance in image recognition and classification. Using this model, the website can identify relevant characteristics in uploaded images and suggest products that are similar to the user's needs. This would result in greater customer satisfaction and loyalty in addition to raising user engagement.

Additionally, this project offers a fascinating chance for innovation and research. The use of deep learning methods, especially the ResNet model, can result in the creation of stronger and more precise image recognition and recommendation systems.

In conclusion, the incorporation of an image search feature and product recommendations based on uploaded photographs utilizing the ResNet model within an e-commerce website presents promising possibilities for innovation and advancement in the field.

VI. PROPOSED METHODOLOGY

In recent years, the usage of image search and recommendation systems on e-commerce websites has grown in popularity. Image-based product recommendation is a technique for recommending products to users based on the visual characteristics of the product photos.

The methodology for product recommendation using ResNet50 involves the following steps:

A. Data Collection: The first step is to gather information about the products that should be recommended. Images and product information, such as name and description, are included in this data.

B. Preprocessing: Preprocessing steps are required before feeding the images to the ResNet50 model. The photos

DOI: 10.5281/zenodo.7949796
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam

are resized to a set size of 224x224 pixels, converted to arrays, and the pixel values are normalised.

C. **Feature Extraction:** The ResNet50 model is used to extract features from images after they have been preprocessed. The model has been pre-trained on a huge dataset of photos and is capable of extracting significant characteristics from them. ResNet50 is a deep neural network design popular for image categorization and feature extraction. To extract features from product photos, we can use the pre-trained ResNet50 model that was built on the ImageNet dataset.

D. **Building Product Feature Database:** Once we've extracted features from all of the product images, we can create a database of product features and their corresponding image filenames. Using the pickle module, we can save this database to a file.

E. **Querying Product Feature Database:** When a user uploads an image, we can extract features from the image using the ResNet50 model. We can then query the product feature database to locate the products that are most comparable to the submitted image based on their features. We can use the k-nearest neighbours algorithm to locate the k most comparable products.

F. **Product Recommendation:** Once we've identified the 'k' most similar products, we can show them to the user as product recommendations.

```python
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()
])

img = image.load_img(newImage.image.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)

neighbors = NearestNeighbors(n_neighbors=5, algorithm='brute', metric='euclidean')
neighbors.fit(feature_list)

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

products = []
```

Fig.1

The [Fig.1] shows a function that accepts a request object as an input parameter. It retrieves an uploaded image from the request object and saves it to the image model. The function loads a pre-trained ResNet50 model from Keras to extract features from the stored image. Because the model was trained on the ImageNet dataset, it is capable of recognising a wide range of objects. The feature vector created from the image is then used to discover the most comparable images using the K-nearest neighbour algorithm with Euclidean distance as the metric. The top 5 related goods are retrieved from the database based on the picture filenames. If the products are located, they are saved in a list and provided to the search.html template to be displayed.

```python
def extract_features(img_path, model):
    img = load_img(img_path, target_size=(224, 224))
    img_array = 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

filenames = []

for file in os.listdir('media/productimages'):
    filenames.append(os.path.join('media/productimages', file))

feature_list = []
for file in tqdm(filenames):
    feature_list.append(extract_features(file, model))
#print(np.array(feature_list).shape)
pickle.dump(feature_list,open('embeddings.pkl','wb'))
# Convert filenames to a list before saving to pickle file
pickle.dump(filenames, open('filenames.pkl', 'wb'))

feature_list=np.array(pickle.load(open('embeddings.pkl','rb')))
# Load filenames as a list from pickle file
filenames = pickle.load(open('filenames.pkl', 'rb'))
products = fetch_product(filenames, feature_list, model,image_path)
return products
```

Fig. 2

The function extract_features in [Fig.2] takes an image path and a pre-trained machine learning model as input and outputs a normalised feature vector for the image. The function first loads the image from the specified path, resizes it to 224x224 pixels, converts it to an array, expands the array dimensions to include a batch dimension of size 1, preprocesses the image data to meet the input requirements of the specified model, and finally runs the image through the model to generate a feature vector. Before returning the feature vector, it is normalised using L2 normalisation.

```python
for file in indices[0][1:6]:
    print(filenames[file])
    filename=filenames[file].split("/")[1].replace("\\","/")
    product = Product.objects.filter(image=filename).first()
    print(filename)
    print(product)
    if product:
        products.append(product)


return products
```

Fig.3

[Fig.3] depicts how main code runs through all files in a directory ('media/productimages'), calls the extract_features function on each file, and then appends the resulting feature vector to a list. This list of feature vectors is then saved to the pickle file 'embeddings.pkl', along with a list of image filenames in 'filenames.pkl'.

DOI: 10.5281/zenodo.7949796
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam

VII. RESULT

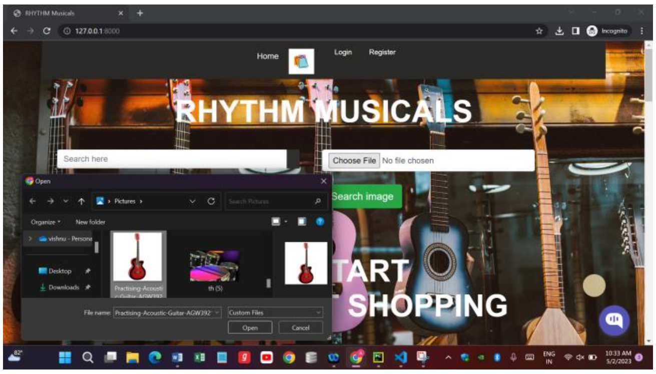

Fig.4

Figure [4] depicts the search done by uploading image from user end and the system takes an image path and a model as input, loads the image and preprocesses it, feeds it to the model, extracts the feature vector, normalizes it, and returns it.

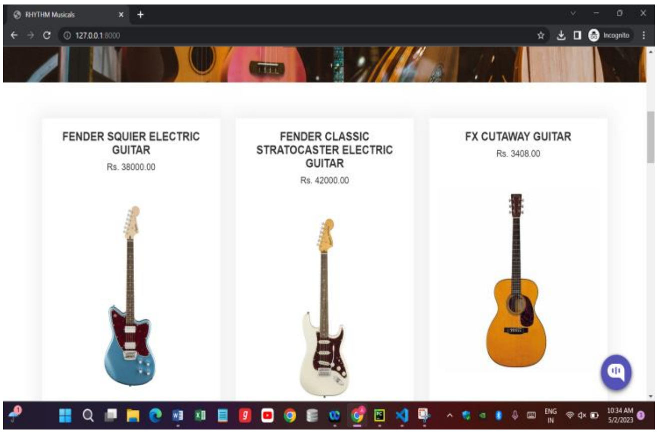

Fig.5

A list of filenames of the images to be processed are created, and then extracts features from each image using the **extract_features** function and stores them in a list called **feature_list**. The **filenames** list is also stored using pickle.

The feature_list and filenames are obtained by loading the embeddings.pkl file. After that, a ResNet50 model is created and utilised to extract features from a query image. After that, the retrieved feature vector is normalised.

The feature_list is used to generate a nearestneighbors object, and the fit method is used to train the model. With the normalised feature vector as input, the kneighbors method is invoked to obtain the indices of the top 6 closest images to the query image.

The loop presents the top five images that are the most similar to the query image. Finally, the similar images are shown in the homepage [Fig.5].

VIII. CONCLUSION

In conclusion, we proposed a ResNet model technique that uses deep learning to recommend items based on their visual similarity. We implemented a system that lets users search for a product by uploading an image, and similar products will be recommended. It extracts characteristics from photos using the ResNet50 deep learning model and then uses the Nearest Neighbours algorithm to locate related images based on Euclidean distance. This method could be used for picture-based product suggestions, where the user enters an image of a product they like and the algorithm discovers related products.

IX.REFERENCES

[1] S. Hussain, M. O. Abo-Zahhad, M. A. Hossain, and M. S. Rahman, "An Image-Based Product Recommendation System using Deep Learning Techniques," IEEE Access, vol. 9, pp. 29873-29885, 2021.

[2] R. Li, X. Chen, Z. Liu, and X. Wang, "A Deep Learning-Based Product Recommendation System with ResNet50 and Attention Mechanism," IEEE Access, vol. 9, pp. 64439-64450, 2021.

[3] Z. Huang, X. He, and J. Chen, "Visual Product Recommendation with Graph Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 3, pp. 915-925, 2021.

[4] M. V. K. S. K. Mahesh, S. Senthil Kumar, and K. Thangaraj, "An Image-Based Product Recommendation System using ResNet50 and Collaborative Filtering Algorithm," in Proceedings of the 2021 International Conference on Data, Engineering and Applications, pp. 159-164, 2021.

[5] S. Vijayakumar, K. Raja, and S. Sathiya, "A Deep Learning-Based Product Recommendation System using ResNet50 and Convolutional Neural Network," in Proceedings of the 2021 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications, pp. 253-259, 2021.

[6] G. Zhang, J. Zhang, Z. Zhang, and Y. Gu, "Visual Search and Recommendation System using Deep Learning and Graph-based Algorithm," in Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Computer Applications, pp. 1-6, 2021.

DOI: 10.5281/zenodo.7949796
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam