[**Introduction to Deep Learning**](https://estuoys.eskisehir.edu.tr/courses/318)

**Term Project**

Abdullah Enes KOCABUGA – 62125459644

Muhammed Hasan GÜNAY - 18410394954

Yusuf Mücahit SOLMAZ – 45394596814

**Problem definition**

The main problem we solved in this project is to reveal the closest estimates using similarities over different images of different people.

Using the image gallery entered as a dataset, giving 1 reference image over different images of 5 registered people, and providing output by guessing the most similar images in the gallery thanks to probability.

**Explanation of the neural network model used**

The main reason we use CNN (Convolutional Neural Network) in this project is because it is an efficient method for image processing and visual feature extraction. CNNs are a kind of artificial neural network that gives successful results especially in deep learning domain image structure, end-of-object and similar tasks.

Our goal in the project is to compare viewpoints with other images in a given query gallery. This drawing calculation includes the visual properties of the image. CNNs are a deep learning model that is particularly suitable for extracting a decor features.

The ResNet V2 model used in this project is a well-known CNN model and TensorFlow Hub is widely used. ResNet V2 enables better feature extraction with its deep mesh structure and generally performs well in image processing tasks.

Using CNN is important to gain a stronger basis for computations to better understand and interpret imaging deficiencies and visual features. Therefore, CNN was used in this project.

**Description of your dataset and how it was obtained**

We obtained our data set from the internet in accordance with our purpose. Our dataset is 224x224 px images of 5 actors from the popular Marvel movie series from different angles. Thanks to the resolution and angular compatibility of the pictures, we have increased the prediction probability of artificial intelligence.

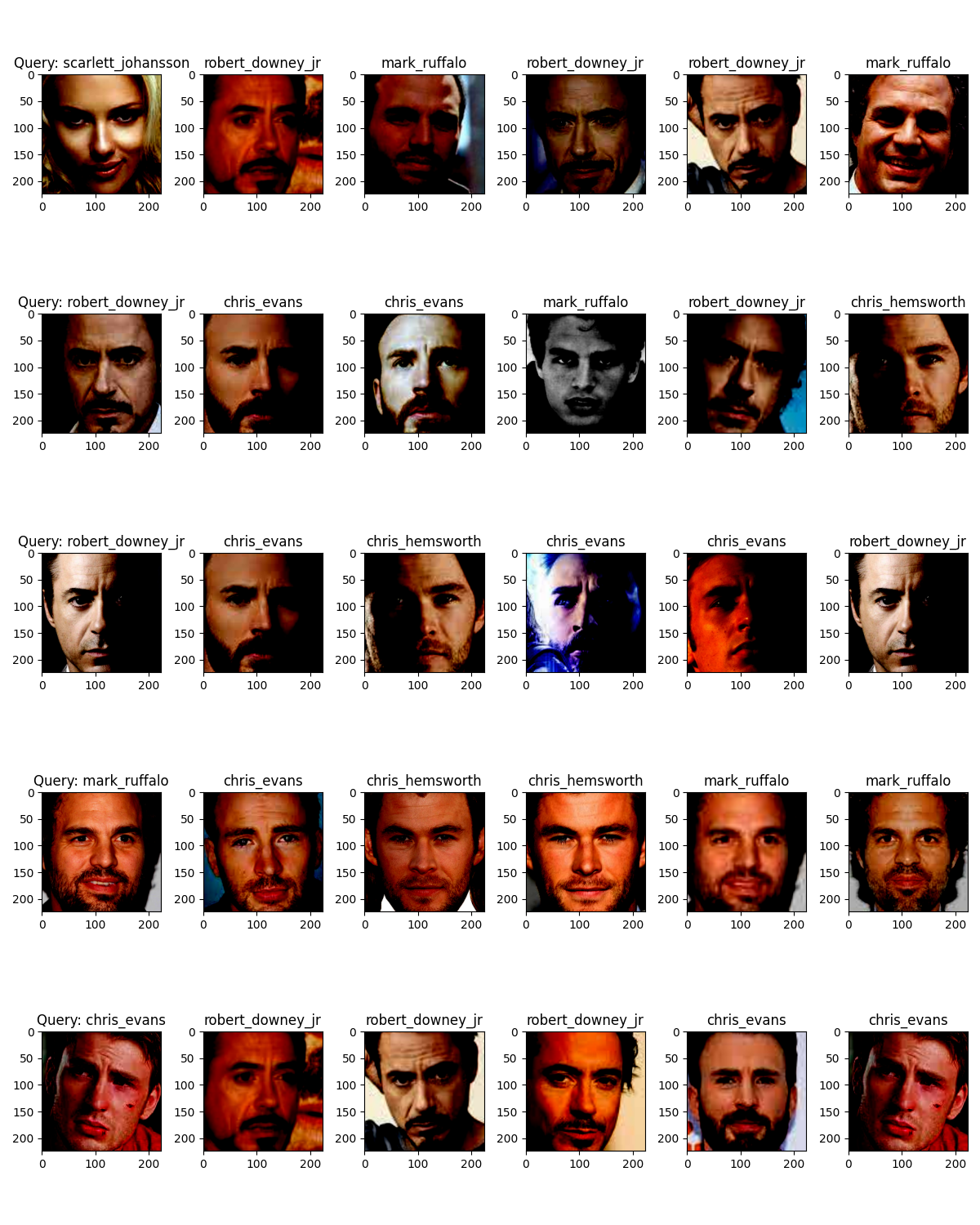
There are 274 images in total from 5 different people in our dataset. In our current project, we randomly print 25 images to use the first 100 images.

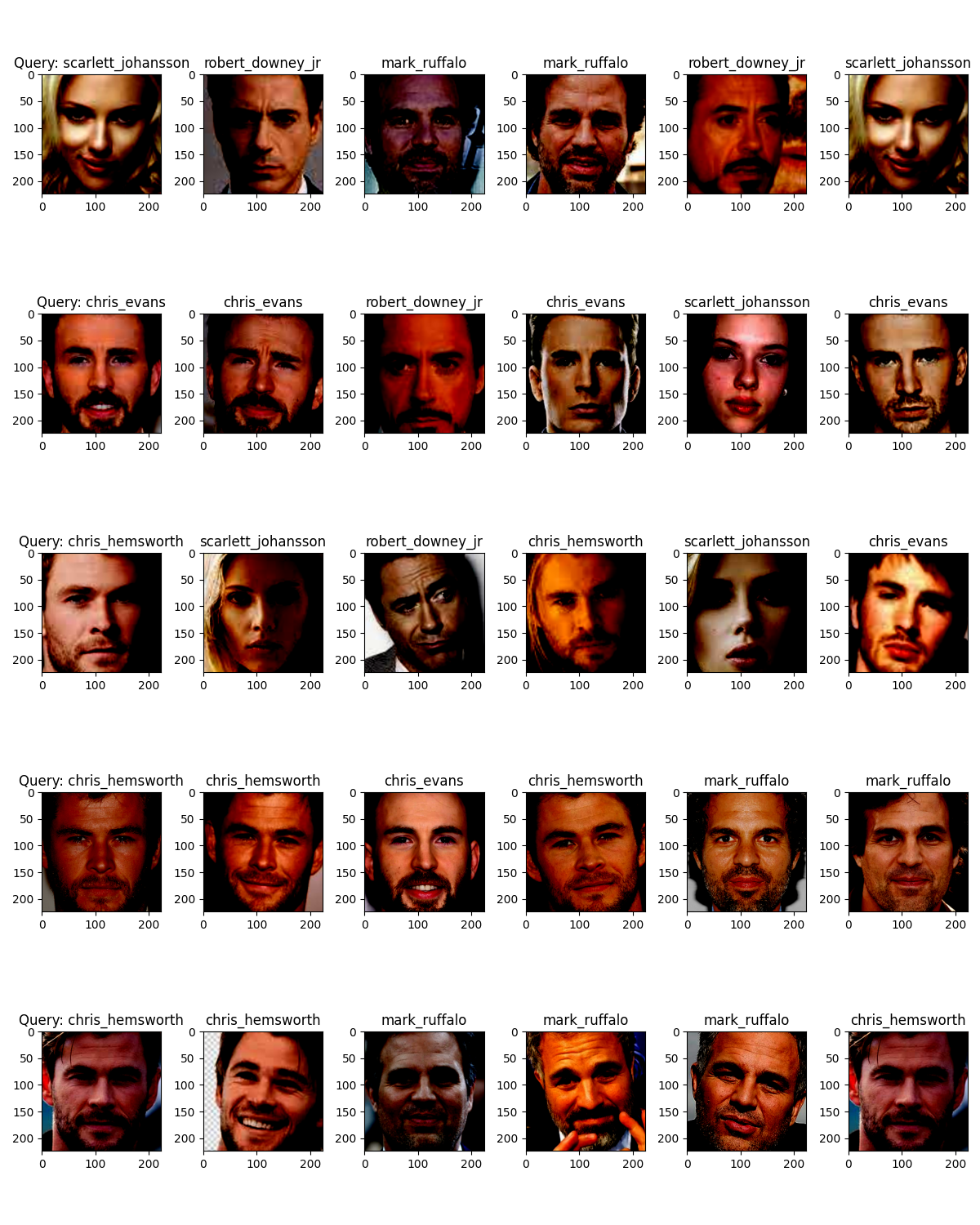
We obtained our data set as an open source internet, which was created especially for use in the field of image processing.

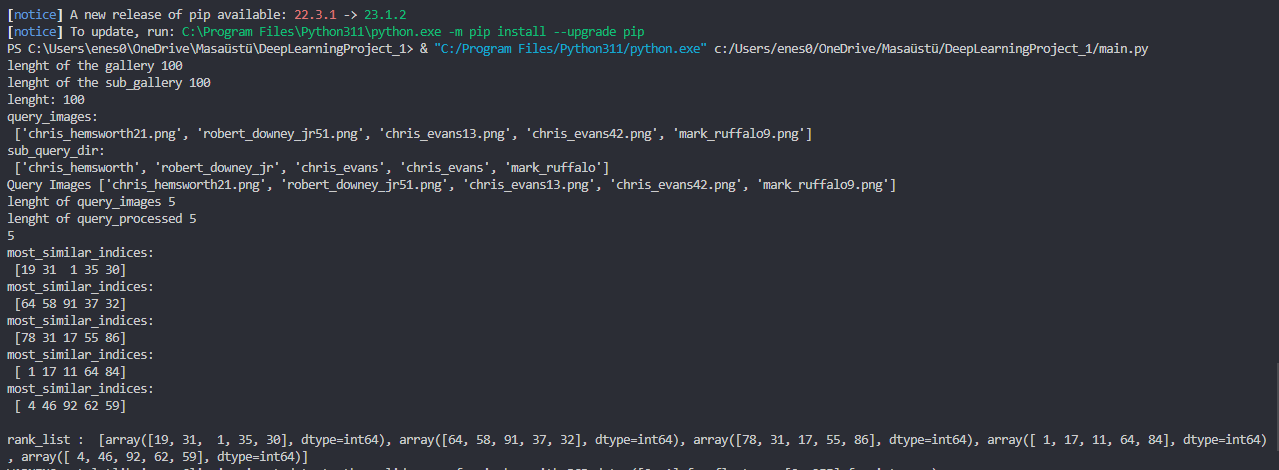
**Results**

In accordance with our purpose, we got the outputs we expected from our project. We designed our code to use 100 images from our dataset containing 274 images. Using 5 reference images each time, we printed out the closest 4 different images in our data set according to the similarity ratio for each. In total, we get 25 images listed according to the similarity ratio as results.

Outputs we get:



****

****

**Discussions**

This code has a structure that performs the operations of creating an image gallery and finding similarity of query images according to this gallery. First, the properties of the gallery and query images are extracted using keras\_layer, the feature extraction layer of the ResNet-152 model.

The gallery creation process starts with the use of randomly selected images from the folders in the specified home folder path. Each image is sized through preprocessing steps and a feature vector is obtained using the ResNet-152 model. These feature vectors are used as representatives of gallery images.

Query images are processed in a similar way. The randomly selected images from the folders are first sized and their feature vectors are obtained. The similarity scores are calculated by comparing the feature vectors of the query images with the feature vectors of the gallery images.

Similarity scores are calculated using the cosine similarity measure. For each query image, the cosine similarity to the feature vectors of the gallery images is calculated and sorted. Gallery images with the highest similarity score are determined as the images with the most similarity.

Finally, the visualization of query images and most similar gallery images is made using the matplotlib library. This visualization allows the user to compare query images with the most similar images in the gallery.

This code provides a search functionality based on image similarity and can be used especially in applications that work with visual data, such as creating image galleries and querying. Image similarity search systems have an important role in many areas such as image-based sample matching, search and classification.

**How to Run**

You can "Run" the libraries imported to the project after they are loaded into the project.

**Source**

<https://www.kaggle.com/datasets>

<https://www.kaggle.com/models>