Reverse Visual Search on LFW Dataset

By
Nguyen Khoa,
Het Shah,
Tobe Yosuke

Reverse Image Search- Baseline

What is reverse image search? Reverse image search is a search engine technology that takes an image file as an input query and returns results related to the image. For this project, the image file was a person's face which would be used to find images that contained the same person. This would result in the use of facial recognition which is a way of identifying or confirming an individual's identity using their face.

Before talking about the specific algorithm we used for this project, some things need to be declared. This includes classifying the images that will be used. This is stated as the model variable. Extracting the images and extensions are needed as well. Lastly, before implementing our algorithm, simply set paths for the data that will be used which are the images to compare. The necessary libraries should be imported as well which can be done while implementing the algorithm.

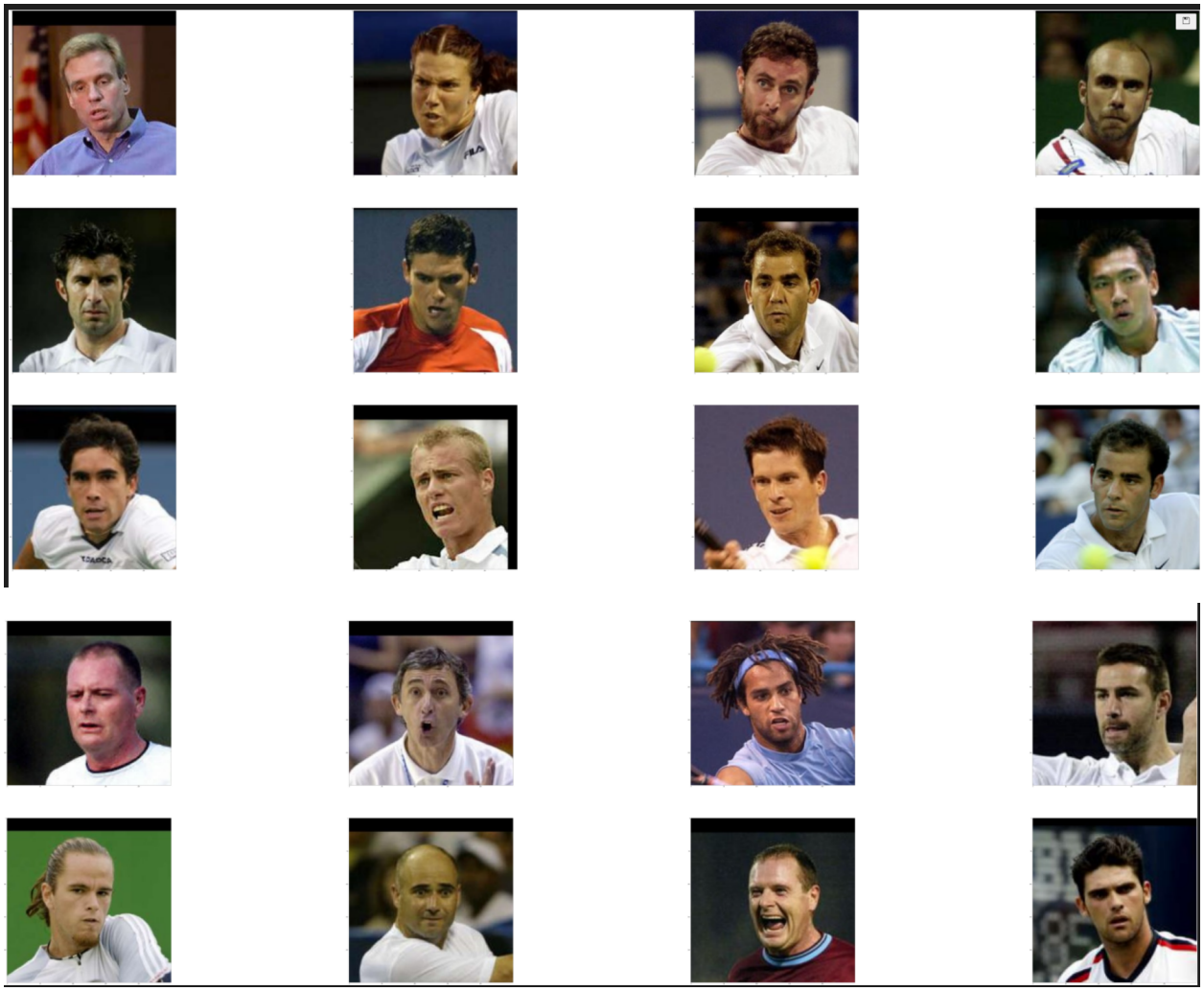
For this project, we used the k-nearest neighbors(KNN) algorithm because it is simple and easy to implement. There is no need to build a model, tune several parameters, or make additional assumptions, and the algorithm is versatile. It can be used for classifications, regression, and search. We used KNN for not only comparing similar images but also to compare similar facial features. For the first test feature, you can see that it is comparing the images that have similar features such as similar facial expressions. In order to apply the KNN classification, we need to define a distance metric and we used the Euclidean distance to compare images for similarity. Below is the image of the result we got from our baseline performance.

```
Median distance between all photos:  0.97451794
Max distance between all photos:  1.0944438
Median distance among most similar photos:  0.969303
```

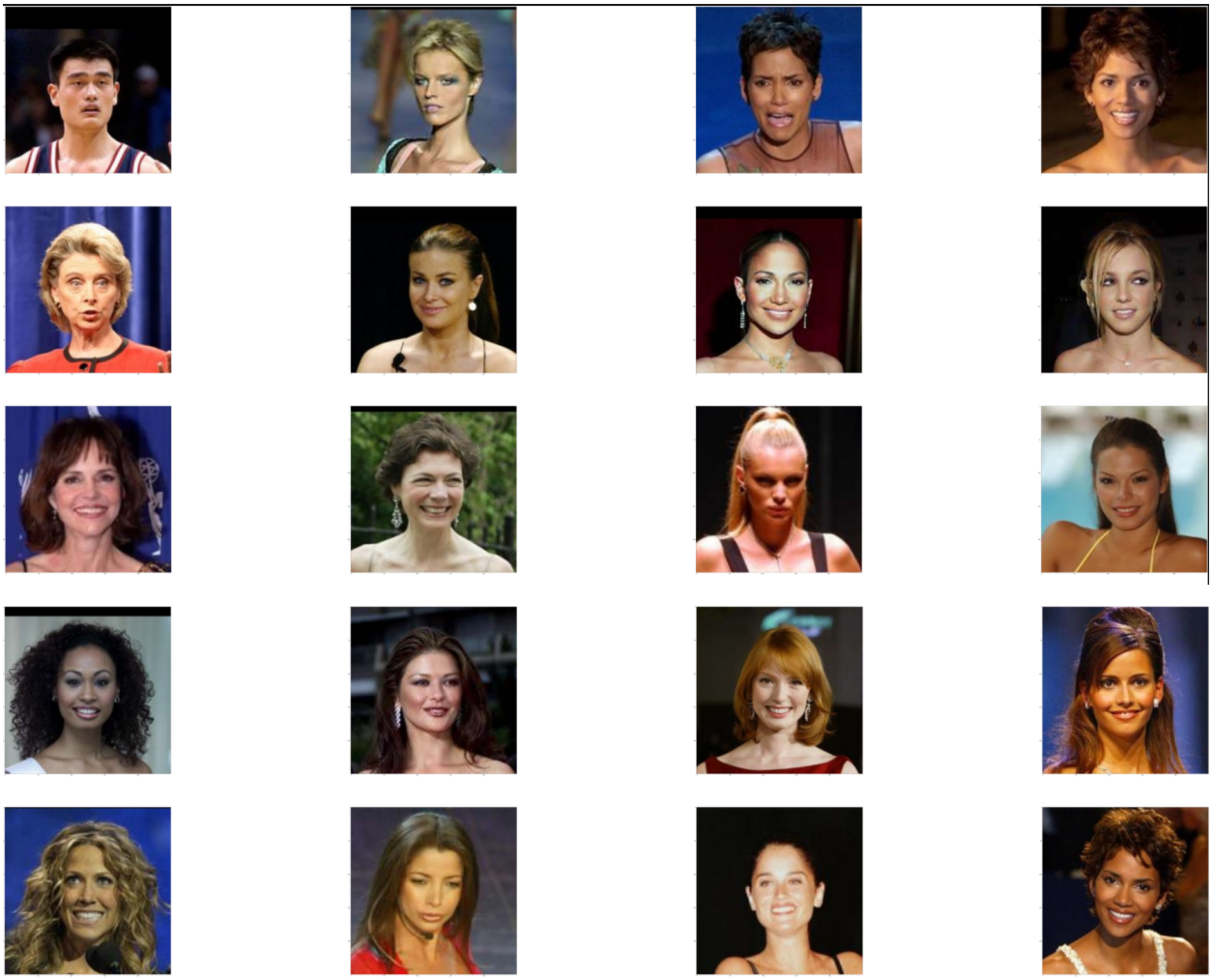
The median distance among most similar photos is 0.969303 which means that the images are not similar to the test case. If the number was lower then it means that the images are similar to the test case according to the Euclidean distance.

Due to the file size of the base case notebook, it could not be uploaded to GitHub and will be shown here instead for the accuracy of the baseline reverse image search. These are some of the similar faces/facial features that we got from our baseline system. Test case examples:

Test_feature[0]



Test_feature[1]



Reverse Search Improvement

Following the system that we created, we used deep face to improve the baseline performance that we obtained earlier. To begin with, what is deepface?. Deep face is a lightweight face recognition and facial attribute analysis (age, gender, emotion, and race) framework for python. It is a hybrid face recognition framework wrapping state-of-art models such as VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, and Dlib.

How to Use/Install DeepFace.

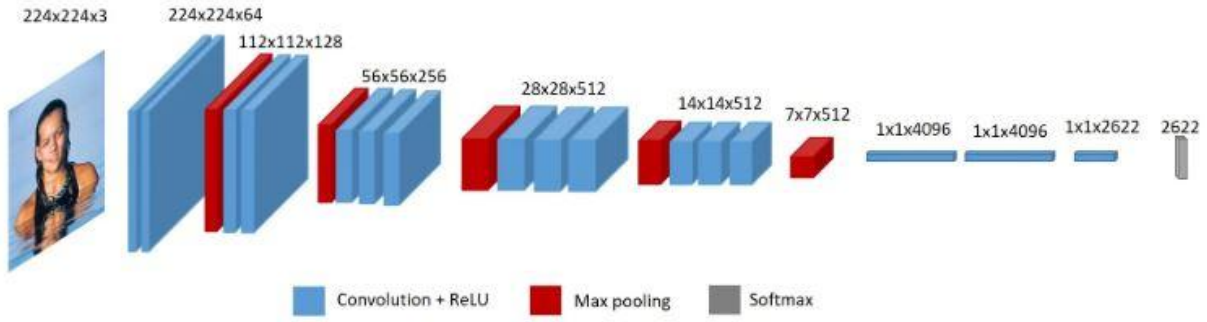
- Open Command Prompt
- Type pip install deepface

Then you will be able to import the library and use its functionalities.

- from deepface import DeepFace

We decided to run deepface on the developed model on the whole dataset using the VGG-Face Model. This method has an accuracy of 98.78%. It has 1 network and has trained over 2.6 million images. Train a convolutional neural network for face recognition. The VGG-Face model has 22 layers and 37 deep units. Below is the structure of the VGG-Face model.

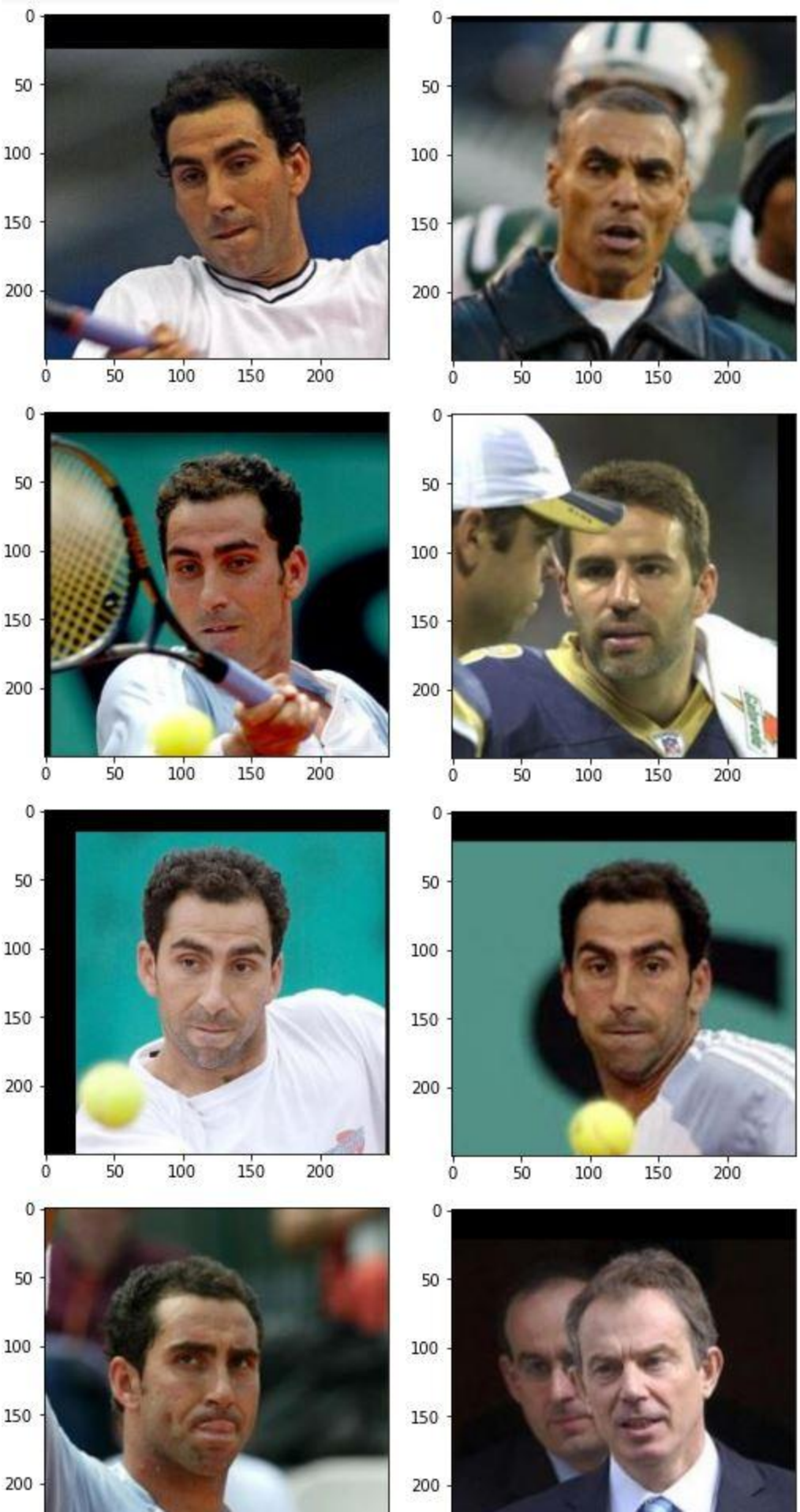
To visualize this structure above on how it works on the image below illustrates how it works.

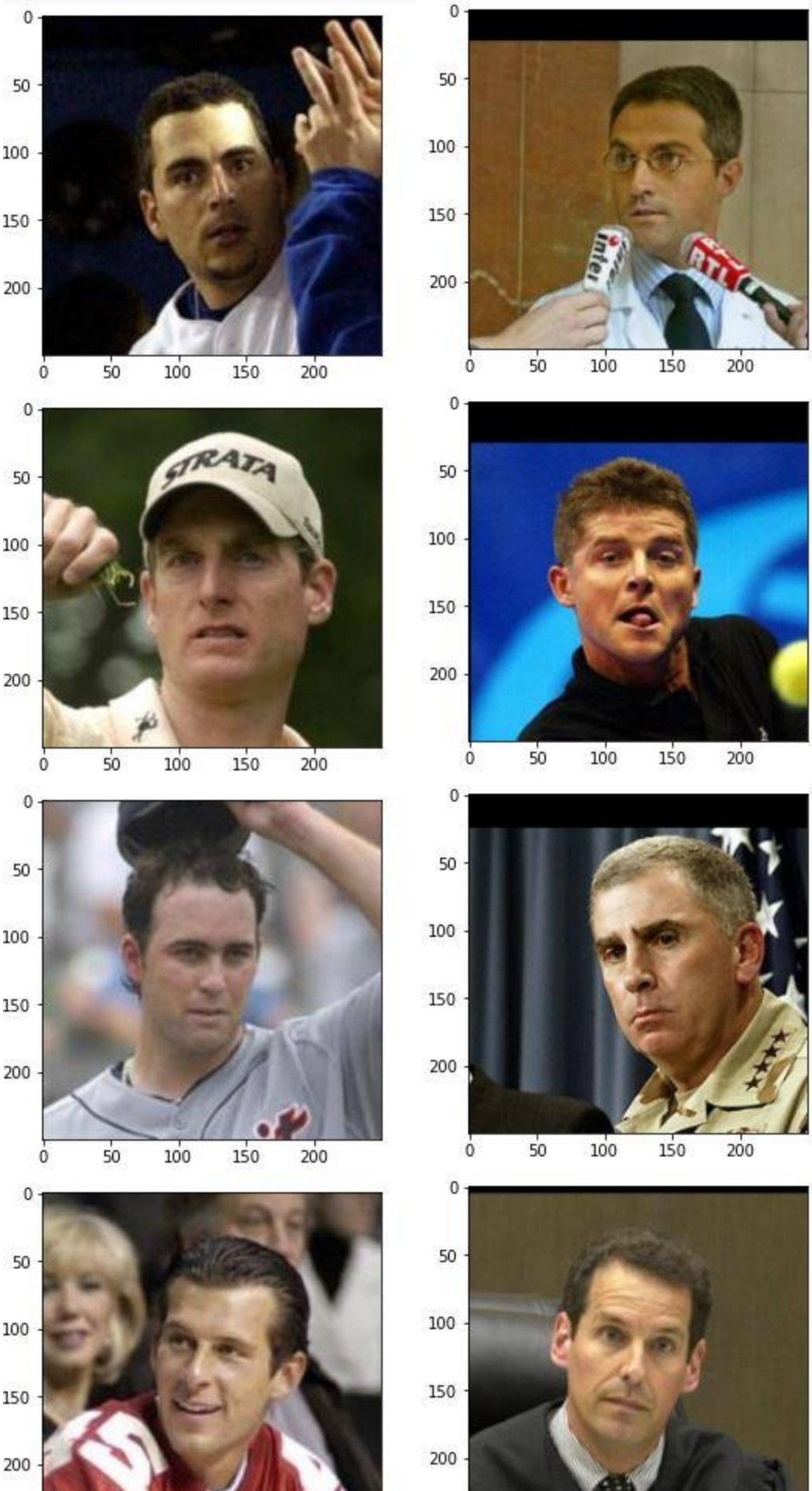


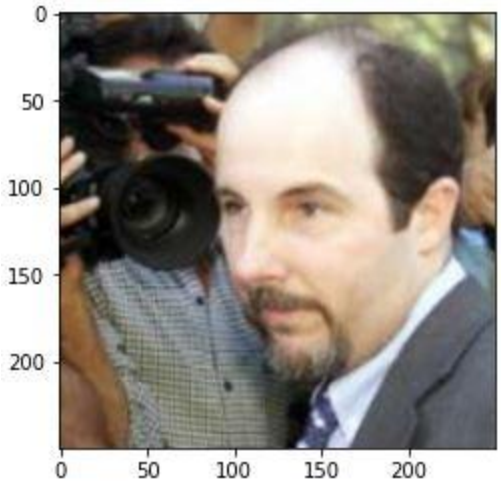
After running the model on the dataset using Deepface we got the following results. Comparing the results that we got from the other system. Meaning using the Euclidean distance when we ran our test, our results were fairly closer to each other than the baseline system. This would result in our improved model being better than the baseline model at reverse image searching. If the number was lower, it means that the images are similar to the test case according to the Euclidean distance.

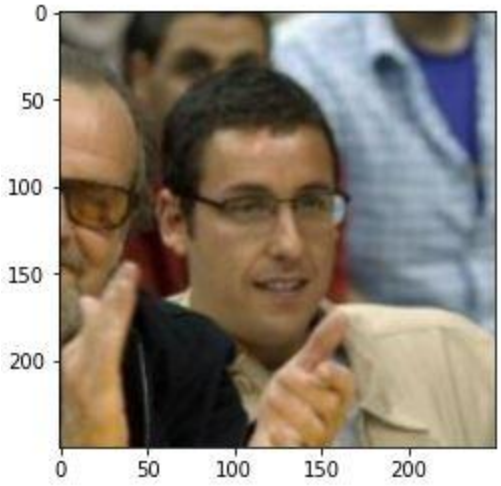
	identity	VGG-Face_cosine
0	data/datasets\Albert_Costa/Albert_Costa_0001.jpg	0.000000
1	data/datasets\Albert_Costa/Albert_Costa_0005.jpg	0.192646
2	data/datasets\Albert_Costa/Albert_Costa_0006.jpg	0.209962
3	data/datasets\Albert_Costa/Albert_Costa_0004.jpg	0.233495
0	data/datasets\Herman_Edwards/Herman_Edwards_00...	0.237015
0	data/datasets\Kurt_Warner/Kurt_Warner_0001.jpg	0.246371
4	data/datasets\Albert_Costa/Albert_Costa_0002.jpg	0.250210
0	data/datasets\Tony_Blair/Tony_Blair_0107.jpg	0.251714
0	data/datasets\Paul_Lo_Duca/Paul_Lo_Duca_0001.jpg	0.255112
0	data/datasets\Jim_Furyk/Jim_Furyk_0001.jpg	0.255906
0	data/datasets\Mark_Redman/Mark_Redman_0001.jpg	0.259254
0	data/datasets\Ryan_Goodman/Ryan_Goodman_0001.jpg	0.261582
0	data/datasets\Stephane_Delajoux/Stephane_Delaj...	0.261659
0	data/datasets\Jiri_Novak/Jiri_Novak_0009.jpg	0.262435
0	data/datasets\John_Abizaid/John_Abizaid_0006.jpg	0.262644
0	data/datasets\Gary_Paer/Gary_Paer_0001.jpg	0.265718
0	data/datasets\Arminio_Fraga/Arminio_Fraga_0002...	0.265795
0	data/datasets\Adam_Sandler/Adam_Sandler_0004.jpg	0.266695
0	data/datasets\Christopher_Amolsch/Christopher_...	0.269391
0	data/datasets\Stephen_Keener/Stephen_Keener_00...	0.270538
0	data/datasets\Steven_Briggs/Steven_Briggs_0001...	0.271988

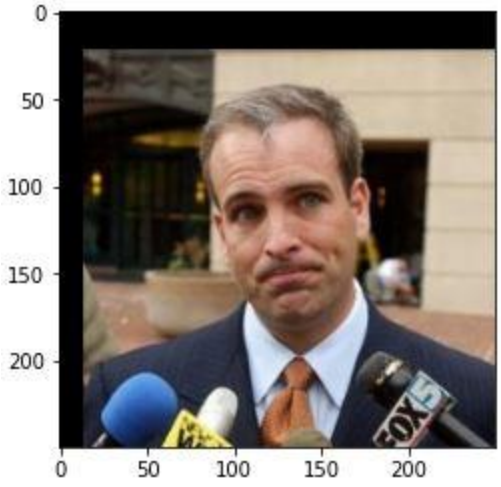
These are the top 21 images from the datasets that are similar to our target image from the datasets.


0
50
100
150
200
0
50
100
150
200

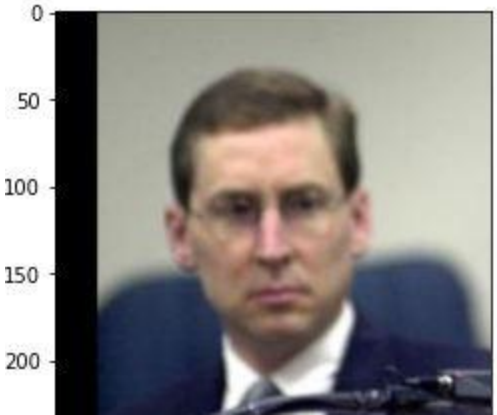

STRATA
RTL


0
50
100
150
200
0
50
100
150
200


0
50
100
150
200
0
50
100
150
200


0
50
100
150
200
0
50
100
150
200


0
50
100
150
200
0
50
100
150
200


0
50
100
150
200