
Pengenalan Wajah dengan Deepface

Deepface adalah suatu framework untuk pengenalan wajah (face recognition) yang membungkus state-of-the-art model-model pengenalan wajah yang telah dikembangkan sebelumnya seperti VGG-face, Google FaceNet, OpenFace, Facebook Deepface, DeepID, ArcFace dan Dlib. Pada tugas ini anda akan mengeksplorasi Deepface Library sehingga mendapatkan prosedur dan konfigurasi terbaik untuk task pengenalan wajah.

Wajah yang akan menjadi referensi utama adalah database pas foto (area citra sebagian besar didominasi wajah) dari seluruh peserta kelas Pembelajaran Mesin Lanjut. Sedang image yang akan dikenali wajahnya (Test Data) berupa foto-foto orang dengan tampilan penuh (setidaknya lebih dari setengah badan) dengan pose yang bebas, tetapi tetap terlihat wajahnya. Setiap foto di Test Data dan database pas foto diberi label sesuai dengan nama yang ada di wajah. Jika foto di Test Data bukan anggota kelas, maka labelnya 'unknown'. Test Data berisi 50% anggota kelas dan 50% non-anggota kelas. Kelas diharapkan bisa berkolaborasi menghasilkan dua dataset tersebut untuk dipakai bersama.

Dengan menggunakan fungsi-fungsi yang disediakan library Deepface, buat prosedur (program) untuk dapat mengenali wajah citra yang ada di Test Data. Selanjutnya lakukan eksplorasi untuk mendapatkan kombinasi model pengenalan wajah dan deteksi wajah dengan kinerja terbaik. Ukuran kinerja yang dipakai adalah akurasi, berdasarkan kemampuannya untuk mengenali wajah-wajah yang ada di Test Data. Pengenalan juga dihitung benar jika wajah yang tidak ada di referensi utama dikategorikan sebagai 'unknown'. Selain menampilkan akurasi di bagian akhir, untuk setiap foto yang ada di Test Data, tampilkan juga hasil prediksinya dengan format sebagai berikut:

Jika sistem berhasil mengenali wajah:

<Foto+Label dari Test Data> <Foto+Label dari Database pas foto> Benar/Salah

Jika sistem tidak berhasil mengenali wajah:

<Foto+Label Test Data> Unknown Benar/Salah

Benar/Salah pada format di atas ditulis berdasarkan penilaian hasil prediksi.

Deliverable berupa file notebook yang berisi program dan file pdf yang berisi output hasil prediksi dari setiap foto yang ada di Test Data (sesuai dengan format di atas), serta nilai akurasinya.

Unggah semua deliverable dan file-file yang diperlukan pada github dan submit URL github ke situs perkuliahan sebelum 18 Mei 2022 pukul 23.59.

Nama Kelompok:
Freddy Artadima Silaban (33221046)
Yulrio Brianorman (33221012)

JAWABAN : PENJABARAN PROGRAM

1. Create Dataset.PKL

A. Instalasi Deepface

```
In [1]: !pip install deepface

Requirement already satisfied: deepface in /usr/local/lib/p
Requirement already satisfied: Flask>=1.1.2 in /usr/local/l
Requirement already satisfied: numpy>=1.14.0 in /usr/local/
Requirement already satisfied: gdown>=3.10.1 in /usr/local/
Requirement already satisfied: Pillow>=5.2.0 in /usr/local/
```

B. Mount Google Drive

```
In [ ]: from google.colab import drive
        drive.mount('/content/drive')

In [ ]: %cd /content/drive/MyDrive/S3/Kuliah/Semester2/EI7007-Pembelajaran Mesin Lanjut/Tugas/Kelompok_2/PML-TugBes2
```

C. Import Library

```
In [ ]: import os
        import cv2
        import pickle
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        from sklearn.metrics.pairwise import cosine_similarity

        from PIL import Image
        from os import listdir
        from deepface import DeepFace
```

D. Memproduksi, Menyimpan, Memuat dan Menampilkan data dari wajah.

1). Memproduksi data dari wajah dengan model Facenet

```
In [ ]: folder='AnggotaKelas/'
        database = {}
        models = ["VGG-Face", "Facenet", "Facenet512", "OpenFace", "DeepFace", "DeepID", "ArcFace", "Dlib"]
        i = 6

        for filename in listdir(folder):
            path = folder + filename
            number = DeepFace.represent(path, model_name=models[i], align=False,
                                      enforce_detection = True)
            number = np.asarray(np.array(number))
            database[filename] = number
```

2). Menyimpan data wajah ke dalam file .pkl

```
[ ]: myfile = open("data.pkl", "wb")
      pickle.dump(database, myfile)
      myfile.close()
```

3). Memuat data wajah dari file .pkl

```
In [ ]: myfile = open("data.pkl", "rb")
        data = pickle.load(myfile)
        myfile.close()
```

4). Menampilkan data wajah

```
In [ ]: data

Out[ ]: {'Adiyasa Nurfalalah.jpg': array([-7.60770887e-02,  1.44262359e-01,  1.32269654e-02, -2.73445696e-01,
      1.67399962e-02,  1.46759935e-02,  5.96235693e-02, -1.75153792e-01,
      -1.34397969e-02, -1.49373576e-01, -7.13461414e-02, -3.15190665e-02,
      -4.84224617e-01, -2.07688008e-02,  1.93008125e-01, -5.80151789e-02,
      -3.57335269e-01, -3.32799315e-01,  1.37532771e-01, -4.13809836e-01,
      -2.86360890e-01,  2.83307694e-02,  2.81092823e-02, -2.63983816e-01,
      -4.66825366e-02,  2.61747152e-01,  1.77105248e-01,  2.72931844e-01,
      7.22358972e-02, -4.65380400e-01,  1.95367903e-01, -1.32310867e-01,
      2.02500641e-01,  4.05628502e-01, -9.64136422e-02, -5.86082697e-01,
      -3.11141640e-01, -3.37554693e-01, -1.69509351e-02,  2.61194617e-01,
      2.64820188e-01,  7.45877177e-02,  8.77604261e-02, -2.64782935e-01,
      -2.65339524e-01,  3.57908964e-01,  4.68837358e-02, -7.08886534e-02,
      -9.18141529e-02,  2.52425820e-02,  1.52534425e-01,  1.44156795e-02,
      -3.98168802e-01, -1.59461141e-01,  2.92724490e-01, -3.89253944e-01,
      2.41920590e-01,  5.41006327e-01, -2.44018644e-01, -4.68106475e-03,
      8.78512934e-02,  3.85775536e-01, -1.13173425e-01,  6.43389374e-02,
      2.35597014e-01, -3.81237924e-01,  1.19713493e-01, -1.28823042e-01,
      -7.05327332e-01,  1.83067828e-01,  3.53956874e-03, -6.99611753e-02,
      7.22963810e-01, -9.68066156e-02, -3.56332511e-01, -2.82533094e-02,
```

2. Face_Recognition.ipynb

A. Instalasi Deepface

```
In [16]: !pip install deepface

Requirement already satisfied: deepface in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: keras>=2.2.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: Flask>=1.1.2 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: numpy>=1.14.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: tensorflow>=2.4.0 in /usr/local/lib/python3.10/dist-packages
```

B. Mount Google Drive

```
In [17]: from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

In [18]: %cd /content/drive/MyDrive/S3/Kuliah/Semester2/EI7007-Pembelajaran Mesin Lanjut/Tugas/Kelompok_2/PML-TugBes2

/content/drive/MyDrive/S3/Kuliah/Semester2/EI7007-Pembelajaran Mesin Lanjut/Tugas/Kelompok_2/PML-TugBes2
```

C. Import Library

```
In [19]: import os
import cv2
import pickle
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics.pairwise import cosine_similarity

from PIL import Image
from os import listdir
from deepface import DeepFace
```

D. Memuat data wajah dari file .pkl

```
In [20]: database = {}
myfile = open("data.pkl", "rb")
database = pickle.load(myfile)
myfile.close()
```

E. Memilih model yang digunakan

Model yang digunakan adalah ArcFace

```
In [21]: models = ["VGG-Face", "Facenet", "Facenet512", "OpenFace", "DeepFace", "DeepID", "ArcFace", "Dlib"]
i = 6
```

F. Fungsi untuk menghitung Euclidean

```
In [22]: def euclidean_distance(source_representation, test_representation):
euclidean_distance = source_representation - test_representation
euclidean_distance = np.sum(np.multiply(euclidean_distance, euclidean_distance))
euclidean_distance = np.sqrt(euclidean_distance)
return euclidean_distance
```

G. Fungsi untuk mencari wajah

```
In [23]: def mencari_wajah(img_path):
index = 0
prediksi = ''
jarak = 10
pic2 = np.array(DeepFace.represent(img_path, model_name=models[i], align=False,
enforce_detection = True))

for key, value in database.items() :
pic1 = value
distance = euclidean_distance(pic1,pic2)
# print(distance)
if(distance < jarak):
prediksi = key
jarak = distance
index += 1
menampilkan_wajah_cocok(img_path, prediksi, jarak)
```

H. Fungsi untuk menampilkan hasil pencarian

```
In [24]: def menampilkan_wajah_cocok(image_path, prediksi, jarak):
threshold = 4.4
if(jarak < threshold):
print (f"Ini anggota kelas bernama : {prediksi.split('.')[0]}-{round(jarak,3)}" )
else :
print ("Ini BUKAN anggota kelas")

img = plt.imread(image_path)
plt.imshow(img)
plt.axis('off')
plt.show()
```

I. Uji coba Foto yang merupakan anggota kelas.

```
In [25]: img_path = 'fotoTest/meza silvana.jpg'
mencari_wajah(img_path)
```

Ini anggota kelas bernama : Meza Silvana-3.356



```
In [26]: img_path = 'fotoTest/meredita susanty.jpg'
mencari_wajah(img_path)
```

Ini anggota kelas bernama : Meredita Susanty-3.34



J. Uji coba Foto yang merupakan bukan anggota kelas.


```
In [27]: img_path = 'fotoTest/sucipto.jpg'
         mencari_wajah(img_path)
```

Ini BUKAN anggota kelas



```
In [28]: %cd /content/drive/MyDrive/S3/Kuliah/Semester2/EI7007-Pembelajaran Mesin Lanjut/Tugas/Kelompok_2/PML-TugBes2/UploadFiles
         from google.colab import files

         uploaded = files.upload()

         for fn in uploaded.keys():
             print('User uploaded file "{name}" with length {length} bytes'.format(
                 name=fn, length=len(uploaded[fn])))

         %cd /content/drive/MyDrive/S3/Kuliah/Semester2/EI7007-Pembelajaran Mesin Lanjut/Tugas/Kelompok_2/PML-TugBes2
         img_path = "UploadFiles/" + fn
         print("-----")
         mencari_wajah(img_path)
```

/content/drive/MyDrive/S3/Kuliah/Semester2/EI7007-Pembelajaran Mesin Lanjut/Tugas/Kelompok_2/PML-TugBes2/UploadFiles
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving galgadot-2.jpg to galgadot-2.jpg

User uploaded file "galgadot-2.jpg" with length 13734 bytes

/content/drive/MyDrive/S3/Kuliah/Semester2/EI7007-Pembelajaran Mesin Lanjut/Tugas/Kelompok_2/PML-TugBes2

Ini BUKAN anggota kelas

