

Laporan Tugas Kelompok 6

Pengenalan Wajah (*Face Recognition*) dengan Google Facenet

Mata Kuliah : EI7007 Pembelajaran Mesin Lanjut

Nama : **Baud Haryo Prananto** (NIM: 33218009) dan **Arief Sartono** (NIM: 33221018)

URL GitHub Arief Sartono:

https://github.com/ariefsartono/pub_mkei7007_deepface

URL GitHub Baud Haryo Prananto:

https://github.com/prananto/aud.prananto/blob/main/Face_Recognition_using_Facenet.ipynb

Catatan: kedua URL GitHub tersebut di atas berisikan coding yang sama.

Bagian 1 – Penjelasan Alur Program

Telah dilakukan eksplorasi terhadap arsitektur Facenet untuk persoalan prediksi (pengenalan) wajah. Dasar kode program menggunakan contoh program dari Google Facenet. Dataset yang dipergunakan berupa foto-foto orang dengan tampilan penuh (setidaknya lebih dari setengah badan) dengan pose yang bebas, tetapi tetap terlihat wajahnya. 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.
- Peserta kelas berkolaborasi menghasilkan dua dataset tersebut untuk dipakai bersama.



Gambar 1. Visualisasi Dataset Training

Gambar 1 menunjukkan dataset training yang telah dibangun secara bersama oleh para peserta kelas. Label direpresentasikan pada nama direktori untuk masing-masing peserta (total peserta: 26 orang), dengan jumlah image yang diunggah per orang bervariasi antara 1 sampai dengan 16 file

image (rata-rata jumlah image terhadap jumlah peserta: $69/26 = 2,7$ file image per peserta). Selain 26 direktori peserta kelas, terdapat 1 direktori bukan peserta kelas dengan label 'unknown' yang berisikan file image non-anggota kelas. Sedangkan dataset testing berisikan formasi direktori yang sama, yaitu 26 direktori peserta dan 1 direktori bukan peserta kelas dengan label 'unknown' yang berisikan file image non-anggota kelas. Pada dataset testing, jumlah image yang diunggah per orang hanya 1 file image saja. Dimensi gambar bervariasi dengan 3 *channel* warna (RGB). Total terdapat 79 gambar dalam direktori *training* dan 27 gambar dalam direktori *testing*.

Rincian tugas:

- Menggunakan fungsi-fungsi yang disediakan library Deepface, buat prosedur (program) untuk dapat mengenali wajah image yang ada di Test Data.
- Melakukan 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.

Berikut profil dari model *deep learning* yang akan dibangun:

- ✓ Persoalan: Klasifikasi
- ✓ Model: Google Facenet
- ✓ Dataset: Dataset yang dibangun oleh peserta kelas
- ✓ Perangkat keras: Google Colab

Step 1 – Load Facenet Model

Model yang akan dibangun berbasis model Keras FaceNet yang telah dikembangkan oleh Hiroki Tanai (2018) yaitu model Inception ResNet v1 dari TensorFlow ke Keras. Terdapat skrip untuk mengonversi model Inception ResNet v1 dari TensorFlow ke Keras, dengan ketersediaan model Keras yang sudah terlatih siap pakai, yang dilatih pada dataset MS-Celeb-1M berupa gambar input berwarna, memiliki nilai piksel yang *whitened* (terstandarisasi pada tiga *channel* RGB), dan memiliki bentuk persegi 160x160 piksel. Model tersebut telah diunduh dan ditempatkan pada *working directory* dengan nama *facenet_keras.h5*.

Step 2 – Deteksi Wajah (Face Detection)

Sebelum melakukan pengenalan wajah (*face recognition*), kita perlu mendeteksi wajah (*face detection*). Deteksi wajah adalah proses menemukan wajah secara otomatis dalam sebuah image foto dan melokalisasinya dengan menggambar kotak pembatas di sekitar luasnya. Pada langkah ini akan digunakan model *deep learning Multi-Task Cascaded Convolutional Neural Network* atau MTCNN, untuk deteksi wajah (menemukan dan mengekstrak wajah dari sebuah image foto) yang telah dikembangkan oleh Iván de Paz Centeno (2016).

Library *mtcnn* digunakan untuk membuat pendeteksi wajah dan mengekstrak wajah untuk digunakan model pendeteksi wajah FaceNet di bagian selanjutnya. Pertama-tama memuat gambar sebagai array NumPy, yang dapat dicapai menggunakan pustaka PIL dan fungsi *open()*. Dilanjut dengan mengonversi gambar ke RGB, untuk berjaga-jaga jika gambar memiliki saluran alfa atau hitam putih. Selanjutnya dibuat *MTCNN face detector class* dan menggunakannya untuk mendeteksi semua wajah dalam foto yang telah dimuat. Hasil dari proses ini adalah daftar kotak pembatas, di mana setiap kotak pembatas mendefinisikan sudut kiri bawah kotak pembatas, serta lebar dan tinggi. Terkadang library akan mengembalikan indeks piksel negatif, dan ini bisa dianggap sebagai *bug*, yang dapat diperbaiki dengan mengambil nilai absolut dari koordinat tersebut. Koordinat ini dapat digunakan untuk mengekstrak wajah. Dengan menggunakan library PIL untuk mengubah ukuran gambar wajah ini ke ukuran yang diperlukan yang sesuai kebutuhan spesifik model yaitu input wajah persegi dengan dimensi 160x160. Secara keseluruhan semua

proses ini dimasukkan kedalam fungsi `extract_face()` akan memuat foto dari nama file yang dimuat dan mengembalikan wajah yang diekstraksi. Diasumsikan bahwa foto tersebut berisi satu wajah dan akan mengembalikan wajah pertama yang terdeteksi. Fungsi ini dapat dipergunakan untuk mengekstrak wajah sesuai kebutuhan di bagian selanjutnya, yaitu input untuk model FaceNet.

Selanjutnya ditambahkan fungsi untuk membaca subdirektori dataset tertentu (di sini dibuat direktori dengan nama 'train' atau 'val'), mengekstrak wajah, dan menyiapkan dataset dengan nama sebagai label keluaran untuk setiap face yang terdeteksi. Fungsi `load_faces()` akan memuat semua wajah ke dalam daftar untuk direktori tertentu, dalam hal ini 'datakelas/train/ariel_sartono/'.

Sedangkan fungsi `load_dataset()` mengambil nama direktori seperti 'datakelas/train/' dan mendeteksi wajah untuk setiap subdirektori (anggota kelas), dengan menetapkan label untuk setiap wajah yang terdeteksi. Proses ini mengembalikan elemen X dan y dari dataset, yang disimpan sebagai satu file array NumPy yang terkompresi melalui fungsi `savez_compressed()`.

Step 3 – Penyematan Wajah (*Face Embedding*)

Face embedding adalah vektor yang merepresentasikan fitur yang diekstraksi dari wajah. Ini kemudian dapat dibandingkan dengan vektor yang dihasilkan untuk wajah lain. Misalnya, vektor lain yang dekat (dengan ukuran tertentu) mungkin merepresentasikan orang yang sama, sedangkan vektor lain yang jauh (dengan ukuran tertentu) mungkin merepresentasikan orang yang berbeda. Model classifier yang ingin dikembangkan akan mengambil input *face embedding* dan memprediksi identitas wajah. Model FaceNet akan menghasilkan penyematan ini untuk gambar wajah tertentu.

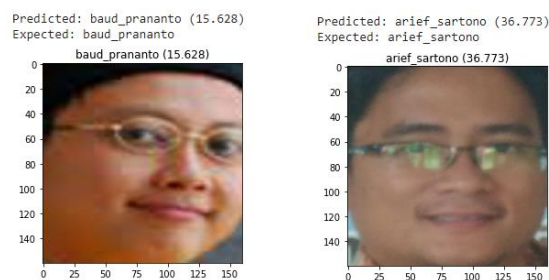
Model FaceNet dapat digunakan sebagai bagian dari pengklasifikasi itu sendiri, atau kita dapat menggunakan model FaceNet untuk melakukan *pre-processing* wajah untuk membuat *face embedding* yang dapat disimpan dan digunakan sebagai input untuk model *classifier* yang akan dibuat. Pendekatan terakhir lebih disukai karena model FaceNet besar dan lambat untuk membuat *face embedding*. Oleh karena itu dilakukan pra-perhitungan *face embedding* untuk semua wajah di

dataset training dan testing (secara resmi 'val') pada dataset. Dataset yang dihasilkan kemudian disimpan ke array NumPy terkompresi dengan nama kelas-embeddings.npz' pada direktori.

Step 4 – Klasifikasi Wajah (*Face Classification*)

Pada bagian ini akan dikembangkan model untuk mengklasifikasikan *face embedding* sebagai salah satu personil yang dikenal pada dataset anggota kelas. Langkah-langkah yang dilakukan antara lain:

- Memuat dataset face embeddings.
- Melakukan beberapa persiapan kecil sebelum pemodelan berupa normalisasi vector, yaitu menskalakan nilai hingga panjang atau besaran vektor adalah 1 atau satuan panjang. Ini dapat dicapai dengan menggunakan class Normalizer pada scikit-learn.
- Variabel target string untuk setiap nama peserta kelas perlu dikonversi ke bilangan bulat. Ini dapat dicapai melalui kelas LabelEncoder pada scikit-learn.
- Menyesuaikan model dengan menggunakan *Linear Support Vector Machine (SVM)* saat bekerja dengan input *face embedding* yang dinormalisasi. Hal ini dikarenakan metode tersebut sangat efektif dalam memisahkan vektor-vektor *face embedding*. Memasukkan SVM linier ke data training dilakukan dengan menggunakan class SVC di scikit-learn dan melakukan *setting* atribut 'kernel' ke 'linier'. Kemudian nilai probabilitas pada saat membuat prediksi dapat dikonfigurasi dengan melakukan *setting* nilai probability' ke 'true'.
- Mengevaluasi model dengan menggunakan fit model untuk membuat prediksi untuk setiap contoh dalam dataset training dan testing, serta menghitung akurasi dari klasifikasi yang dilakukan.



Gambar 2. Hasil Prediksi Wajah

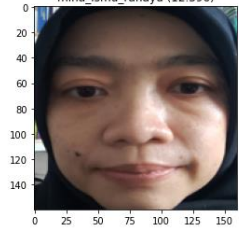

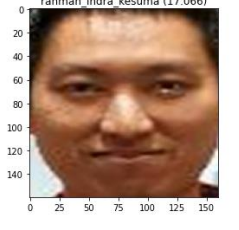

Contoh foto Baud Pranoto dan Arief Sartono dipilih dan diprediksi dengan benar. Gambar 2 menunjukkan plot wajah yang dipilih, dengan menunjukkan nama dan probabilitas yang diprediksi dalam judul gambar.

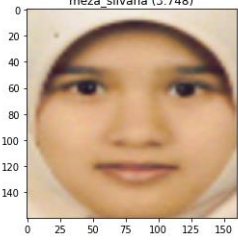

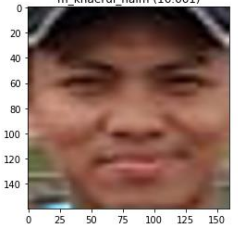

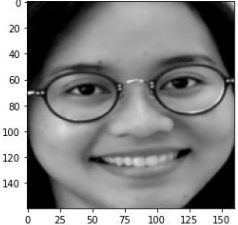


Logic face recognition ini di-looping terhadap dataset *testing* dengan mengklasifikasikan *face embedding* sebagai semua personil pada dataset anggota kelas maupun non-anggota kelas.


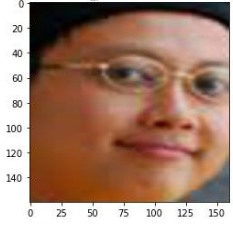

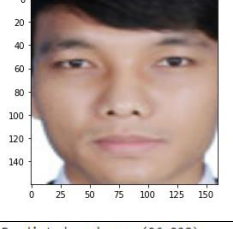
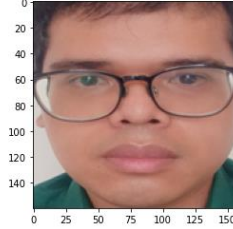
Bagian 2 – Hasil (*Output*) Model

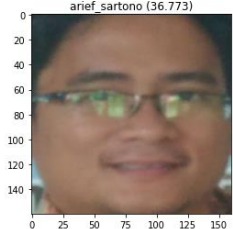

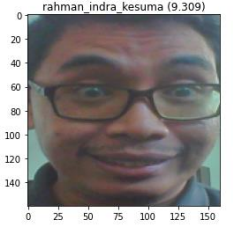
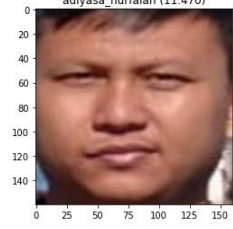

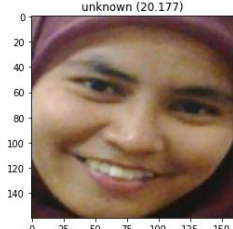
Hasil (*output*) dari eksekusi model yang telah dikembangkan sebagaimana ditampilkan pada tabel 1. Gambar yang disajikan pada foto test data menunjukkan plot wajah yang dipilih, dengan nama (nama hasil prediksi disandingkan dengan *real value* dari label nama direktori) dan nilai probabilitas yang diprediksi pada bagian atas foto tersebut.







Tabel 1. Hasil (Ouput) Model Google Facenet

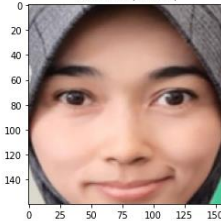
No	Test Data		Training Data		Benar / Salah? **	Anggota / Non-Anggota?	Berhasil Mengenali?
	Foto	Label *	Foto ***	Label *			
1	Predicted: mina_ismu_rahayu (12.396) Expected: mina_ismu_rahayu 	mina_ismu_rahayu (1)		mina_ismu_rahayu (3)	Benar	Anggota	Berhasil
2	Predicted: rahman_indra_kesuma (17.066) Expected: kemas_muhammad_irsan_riza 	kemas_muhammad_irsan_riza (1)		kemas_muhammad_irsan_riza (1)	Salah	Anggota	Tidak Berhasil

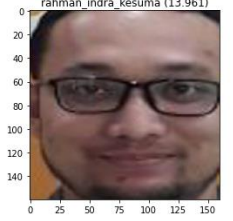

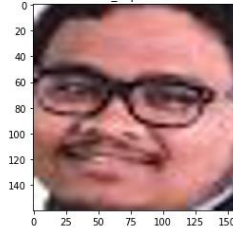

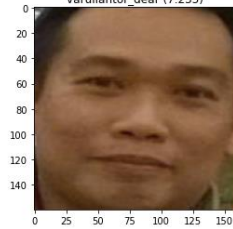

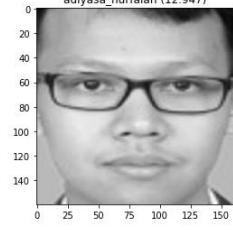
3	Predicted: meza_silvana (3.748) Expected: meza_silvana 	meza_silvana (1)		meza_silvana (2)	Benar	Anggota	Berhasil
4	Predicted: m_khaerul_naim (16.661) Expected: m_khaerul_naim 	m_khaerul_ naim (1)		m_khaerul_ naim (3)	Benar	Anggota	Berhasil
5	Predicted: unknown (17.099) Expected: meredita_susanty 	meredita_ susanty (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil
6	Predicted: yaya_setiadi (3.295) Expected: yaya_setiadi 	yaya_setiadi (1)		yaya_setiadi (2)	Benar	Anggota	Berhasil

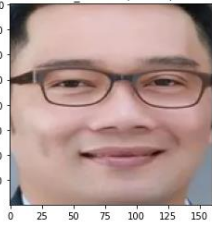
7	Predicted: mina_ismu_rahayu (10.311) Expected: leni_fitriani 	leni_fitriani (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil
8	Predicted: baud_prananto (15.628) Expected: baud_prananto 	baud_prananto (1)		baud_prananto (7)	Benar	Anggota	Berhasil
9	Predicted: rahman_indra_kesuma (12.512) Expected: arief_ichwani 	arief_ichwani (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil
10	Predicted: unknown (26.028) Expected: reza_budiawan 	reza_budiawan (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil

11	Predicted: arief_sartono (36.773) Expected: arief_sartono 	arief_sartono (1)		arief_sartono (16)	Benar	Anggota	Berhasil
12	Predicted: rahman_indra_kesuma (9.309) Expected: riyanto 	riyanto (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil
13	Predicted: adiyasa_nurfalah (11.470) Expected: adiyasa_nurfalah 	adiyasa_ nurfalah (1)		adiyasa_ nurfalah (4)	Benar	Anggota	Berhasil
14	Predicted: unknown (20.177) Expected: lathifah_arief 	lathifah_arief (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil

15	Predicted: unknown (29.789) Expected: sulthoni_ashiddiqi unknown (29.789) 	sulthoni_ ashiddiqi (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil
16	Predicted: imam_ekowicaksono (3.069) Expected: imam_ekowicaksono imam_ekowicaksono (3.069) 	imam_ ekowicaksono (1)		imam_ ekowicaksono (1)	Benar	Anggota	Berhasil
17	Predicted: ahmad_luky_ramdani (3.675) Expected: ahmad_luky_ramdani ahmad_luky_ramdani (3.675) 	ahmad_luky_ ramdani (1)		ahmad_luky_ ramdani (2)	Benar	Anggota	Berhasil
18	Predicted: unknown (19.780) Expected: mohamad_idris unknown (19.780) 	mohamad_idris (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil

19	Predicted: unknown (32.602) Expected: ricky_isfandiani unknown (32.602) 	ricky_isfandiani (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil
20	Predicted: unknown (18.906) Expected: yulrio_brianorman unknown (18.906) 	yulrio_ brianorman (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil
21	Predicted: unknown (17.980) Expected: hartanto_tantriawan unknown (17.980) 	hartanto_ tantriawan (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil
22	Predicted: unknown (25.626) Expected: dewi_tresnawati unknown (25.626) 	dewi_ tresnawati (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil

23	Predicted: rahman_indra_kesuma (13.961) Expected: rahman_indra_kesuma 	rahman_indra_ kesuma (1)		rahman_indra_ kesuma (5)	Benar	Anggota	Berhasil
24	Predicted: handoko_supeno (13.430) Expected: handoko_supeno 	handoko_ supeno (1)		handoko_ supeno (4)	Benar	Anggota	Berhasil
25	Predicted: varuliantor_dear (7.235) Expected: varuliantor_dear 	varuliantor_ dear (1)		varuliantor_ dear (3)	Benar	Anggota	Berhasil
26	Predicted: adiyasa_nurfalah (12.947) Expected: achmad_indra_aulia 	achmad_indra_ aulia (1)	unknown	unknown	Salah	Anggota	Tidak Berhasil

27	Predicted: arief_sartono (13.971) Expected: unknown 	ridwan_kamil (1)	unknown	unknown	Salah	Non-Anggota	Tidak Berhasil
----	---	---------------------	---------	---------	-------	-------------	----------------

* Angka di dalam kurung menunjukkan jumlah file image pada direktori (train/val) dari personil yang bersangkutan.

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

*** Foto yang disajikan pada tabel bersifat perwakilan (mengambil salah satu dari file image yang ada)

Bagian 3 – Catatan dan Kesimpulan

Beberapa catatan dari model yang telah dikembangkan pada tugas kelompok ini:

- Model Keras yang sudah terlatih siap pakai (*pre-trained model*), yang dilatih pada dataset MS-Celeb-1M masih berbasiskan selebriti luar negeri dan masih belum berbasiskan anggota kelas.
- Porsi anggota kelas dan non-anggota kelas belum mencapai porsi 50:50.
- Jumlah image foto yang diunggah peserta kelas tidak sama antara peserta satu dengan lainnya.
- Makin banyak data training, makin tinggi nilai akurasi yang dihasilkan.
- Bila terdapat lebih dari 1 (satu) wajah pada gambar yang sama, terdapat kemungkinan salah deteksi wajah.

Kesimpulan dari kondisi di atas adalah, hal-hal tersebut di atas mengakibatkan rendahnya nilai akurasi yang dihasilkan, dan perlu pengembangan *pre-trained model* yang dibuat berdasarkan database anggota kelas.