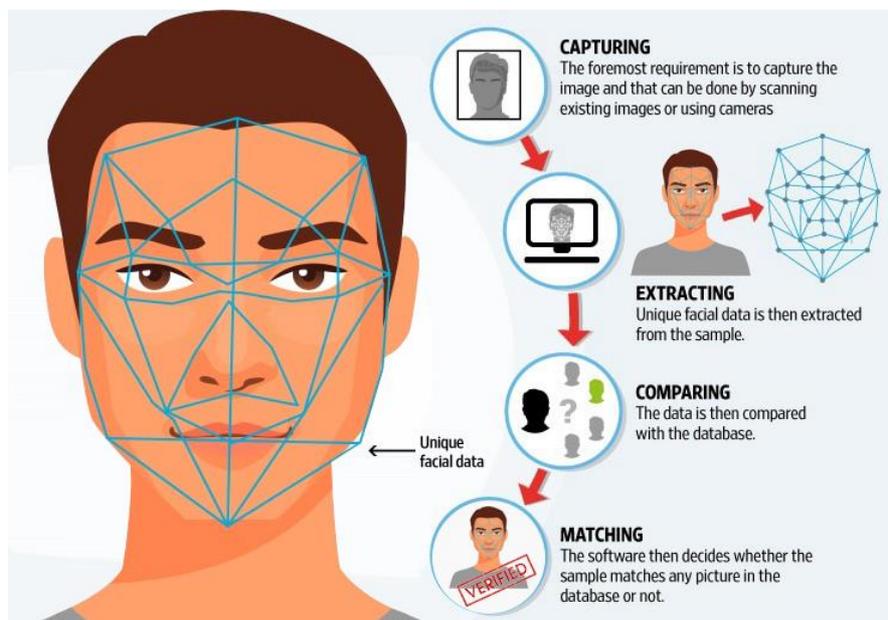


Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri
Aplikasi Nilai Eigen dan EigenFace pada Pengenalan Wajah (*Face Recognition*)
Semester I Tahun 2022/2023

ABSTRAKSI

Pengenalan wajah (*Face Recognition*) adalah teknologi biometrik yang bisa dipakai untuk mengidentifikasi wajah seseorang untuk berbagai kepentingan khususnya keamanan. Program pengenalan wajah melibatkan kumpulan citra wajah yang sudah disimpan pada database lalu berdasarkan kumpulan citra wajah tersebut, program dapat mempelajari bentuk wajah lalu mencocokkan antara kumpulan citra wajah yang sudah dipelajari dengan citra yang akan diidentifikasi. Alur proses sebuah sistem pengenalan wajah diperlihatkan pada Gambar 1.



Gambar 1. Alur proses di dalam sistem pengenalan wajah (Sumber: <https://www.shadowsystem.com/page/20>)

Terdapat berbagai teknik untuk memeriksa citra wajah dari kumpulan citra yang sudah diketahui seperti jarak Euclidean dan *cosine similarity*, principal component analysis (PCA), serta Eigenface. Pada Tugas ini, akan dibuat sebuah program pengenalan wajah menggunakan Eigenface.

Sekumpulan citra wajah akan digunakan dengan representasi matriks. Dari representasi matriks tersebut akan dihitung sebuah matriks Eigenface. Program pengenalan wajah dapat dibagi menjadi 2 tahap berbeda yaitu tahap *training* dan pencocokkan. Pada tahap *training*, akan diberikan kumpulan data set berupa citra wajah. Citra wajah tersebut akan dinormalisasi dari RGB ke Grayscale (matriks), hasil normalisasi akan digunakan dalam perhitungan eigenface. Seperti namanya, matriks eigenface menggunakan eigenvector dalam pembentukannya. Berikut merupakan langkah rinci dalam pembentukan eigenface.

Algoritma Eigenface

1. Langkah pertama adalah menyiapkan data dengan membuat suatu himpunan S yang terdiri dari seluruh training image, ($\Gamma_1, \Gamma_2, \dots, \Gamma_M$)

$$S = (\Gamma_1, \Gamma_2, \dots, \Gamma_M) \quad (1)$$

2. Langkah kedua adalah ambil nilai tengah atau mean (Ψ)

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2)$$

3. Langkah ketiga kemudian cari selisih (Φ) antara nilai training image (Γ_i) dengan nilai tengah (Ψ)

$$\phi_i = \Gamma_i - \Psi \quad (3)$$

4. Langkah keempat adalah menghitung nilai matriks kovarian (C)

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \quad (4)$$

$$L = A^T A \quad L = \phi_m^T \phi_n$$

5. Langkah kelima menghitung eigenvalue (λ) dan eigenvector (v) dari matriks kovarian (C)

$$C \times v_i = \lambda_i \times v_i \quad (5)$$

6. Langkah keenam, setelah eigenvector (v) diperoleh, maka eigenface (μ) dapat dicari dengan:

$$\mu_i = \sum_{k=1}^M v_{ik} \phi_k \quad (6)$$

$$l = 1, \dots, M$$

Tahapan Pengenalan wajah :

1. Sebuah image wajah baru atau test face (Γ_{new}) akan dicoba untuk dikenali, pertama terapkan cara pada tahapan pertama perhitungan eigenface untuk mendapatkan nilai eigen dari image tersebut.

$$\mu_{new} = v \times \Gamma_{new} - \Psi \quad (7)$$

$$\Omega = \mu_1, \mu_2, \dots, \mu_M$$

2. Gunakan metode euclidean distance untuk mencari jarak (distance) terpendek antara nilai eigen dari training image dalam database dengan nilai eigen dari image testface.

$$\varepsilon_k = \Omega - \Omega_k \quad (8)$$

Pada tahapan akhir, akan ditemui gambar dengan euclidean distance paling kecil maka gambar tersebut yang dikenali oleh program paling menyerupai test face selama nilai kemiripan di bawah suatu nilai batas. Jika nilai minimum di atas nilai batas maka dapat dikatakan tidak terdapat citra wajah yang mirip dengan test face.

Program dibuat dengan Bahasa Python dengan memanfaatkan sejumlah library di OpenCV (Computer Vision) atau library pemrosesan gambar lainnya (contoh PIL). Fungsi untuk mengekstraksi fitur dari sebuah citra wajah tidak perlu anda buat lagi, tetapi menggunakan fungsi ekstraksi yang sudah tersedia di dalam library. Fungsi Eigen dilarang import dari library dan harus diimplementasikan, sedangkan untuk operasi matriks lainnya silahkan menggunakan library.

Kode program untuk ekstraksi fitur dapat dibaca pada artikel ini: *Feature extraction and similar image search with OpenCV for newbies*, pada laman:

<https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774>

Nilai batas kemiripan citra test face dapat ditentukan oleh pembuat program melalui percobaan.

Berikut merupakan referensi pengenalan wajah dengan metode *eigenface*:

<https://jurnal.untan.ac.id/index.php/jcskommipa/article/download/9727/9500>

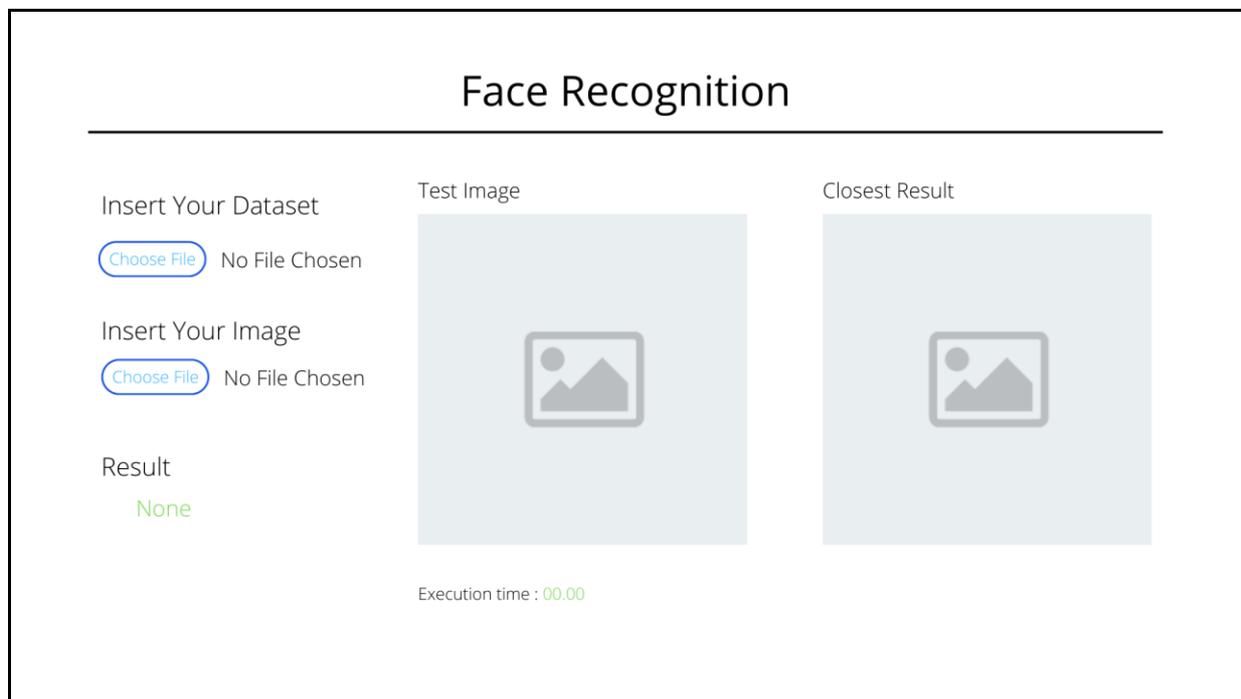
<https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/>

PENGGUNAAN PROGRAM

Berikut ini adalah input yang akan dimasukkan pengguna untuk eksekusi program.

1. **Folder dataset**, berisi *folder* atau *directory* yang berisi kumpulan gambar yang digunakan sebagai *training image*.
2. **File gambar**, berisi *file* gambar input yang ingin dikenali dengan format *file* yang bebas selama merupakan format untuk gambar.

Tampilan *layout* dari aplikasi web yang akan dibangun kurang lebih adalah sebagai berikut. Anda dapat mengubah *layout* selama *layout* masih terdiri dari komponen yang sama.



Gambar 3. Contoh tampilan layout dari aplikasi web yang dibangun.

Catatan: Warna biru menunjukkan komponen yang dapat di klik. Warna hijau menunjukkan luaran yang didapat dari hasil eksekusi.

Anda dapat menambahkan menu lainnya, gambar, logo, dan sebagainya. Tampilan GUI dibuat semenarik mungkin selama mencakup seluruh informasi pada layout yang diberikan di atas. Kreativitas menjadi salah satu komponen penilaian.

SPEKIFIKASI TUGAS

Buatlah program pengenalan wajah dalam Bahasa Python berbasis GUI dengan spesifikasi sebagai berikut:

1. Program menerima input *folder dataset* dan sebuah gambar citra wajah.
2. Basis data wajah dapat diunduh secara mandiri melalui <https://www.kaggle.com/datasets/hereisburak/pins-face-recognition>.
3. Program menampilkan gambar citra wajah yang dipilih oleh pengguna.
4. Program melakukan pencocokan wajah dengan koleksi wajah yang ada di folder yang telah dipilih. Metrik untuk pengukuran kemiripan menggunakan *eigenface* + jarak *euclidean*.
5. Program menampilkan 1 hasil pencocokan pada dataset yang paling dekat dengan gambar input atau memberikan pesan jika tidak didapatkan hasil yang sesuai.
6. Program menghitung jarak *euclidean* dan nilai *eigen* & vektor *eigen* yang ditulis sendiri. Tidak boleh menggunakan fungsi yang sudah tersedia di dalam *library* atau Bahasa Python.

Bonus:

Bagian A (Kamera)

1. Terdapat fitur kamera yang dapat mengenali wajah secara *realtime* menggunakan *webcam* ketika program dijalankan.
2. Teknis pengenalan wajah melalui kamera dibebaskan oleh pembuat program. (contoh: program dieksekusi setiap 10 detik sekali).
3. Fitur kamera merupakan fitur **tambahan**, fitur utama upload gambar melalui GUI tetap harus ada.

Bagian B (Video)

1. Video penjelasan algoritma dan aplikasi program yang diunggah ke youtube.
2. Video dibuat sekreatif mungkin dengan target untuk mensosialisasikan ilmu yang kalian sudah pelajari dan terapkan pada program ini. Bukan hanya video mengenai penggunaan aplikasi.

PROSEDUR Pengerjaan

1. Tugas dikerjakan secara berkelompok yang terdiri dari **maksimal 3 orang** dan **tidak boleh sama dengan anggota kelompok tubes sebelumnya**. Kelompok dipilih secara mandiri dan anggota kelompok diperbolehkan lintas kelas.
2. Daftarkan kelompok anda via tautan <https://bit.ly/DataTubes2AlgeoXXII> sebelum hari **Kamis, 3 November 2022 pukul 21.23 WIB**. Peserta yang tidak terdaftar setelah batas waktu yang ditentukan akan dikelompokkan secara acak oleh tim asisten.
3. Tugas ini dikumpulkan hari **Selasa, 22 November 2022** paling lambat pukul **23.59.59 WIB**. Asisten akan mengumumkan jadwal demo program secara daring.
4. **Dilarang keras menyalin program dari sumber lain (buku, internet, program kakak tingkat, program kelompok lain)**
5. Apabila ada pertanyaan yang ingin ditanyakan, diharapkan untuk bertanya melalui tautan <https://bit.ly/DataTubes2AlgeoXXII>

LAPORAN

Laporan terdiri dari:

1. *Cover*: *Cover* laporan ada foto anggota kelompok (foto bertiga kalau ada, atau foto masing-masing, bebas gaya). Foto ini menggantikan logo “gajah” ganesha.
2. Bab 1: Deskripsi masalah (dapat meng-*copy paste* spesifikasi tugas ini).
3. Bab 2: Teori singkat mengenai perkalian matriks, nilai eigen, vektor eigen, dan eigenface.
4. Bab 3: Implementasi program meliputi penjelasan *tech stack*/kakas yang digunakan dan garis besar algoritma kompresi yang diimplementasikan.
5. Bab 4: Eksperimen, bab ini berisi hasil eksekusi program terhadap contoh-contoh kasus berikut analisis hasil eksekusi tersebut.
6. Bab 5: Kesimpulan, saran, dan refleksi (hasil yang dicapai, saran pengembangan, dan refleksi anda terhadap tugas ini).
7. Daftar Referensi : Referensi (buku, web) yang dipakai/diacu dalam pengerjaan tugas besar.
8. Lampiran: Link repository program dan Link video demo (youtube).

Keterangan laporan:

- a) Laporan ditulis dalam bahasa Indonesia yang baik dan benar, tidak perlu panjang tetapi tepat sasaran dan jelas.
- b) Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

PENGUMPULAN TUGAS

1. Program disimpan di dalam *repository github* dengan nama repository **Algeo02-XXXXX**. XXXXX adalah lima digit terakhir NIM anggota terkecil. Didalam *repository* tersebut terdapat empat folder: *src*, *test* dan *doc* yang masing-masing berisi:
 - Folder *src* berisi *source code*
 - Folder *test* berisi gambar masukan untuk pengujian
 - Folder *doc* berisi laporan dengan format PDFSertakan juga *readme* yang dibuat sebaik mungkin. Pastikan pada *readme* terdapat teknologi yang digunakan serta cara instalasi dan penggunaan website. Silahkan gunakan template <https://github.com/ritaly/README-cheatsheet> atau template lain sebagai referensi. Pastikan juga *repository* bersifat *private* dan telah mengundang asisten yang pembagiannya akan diumumkan kemudian.
2. Format penamaan folder program adalah sebagai berikut : **Algeo02-XXXXX** dengan XXXXX adalah lima digit terakhir NIM anggota terkecil.
3. Format penamaan laporan adalah sebagai berikut: **Algeo02-XXXXX.pdf** dengan XXXXX adalah lima digit terakhir NIM anggota terkecil
4. Folder tersebut di-zip dengan format yang sama dengan nama folder, yaitu **Algeo02-XXXXX.zip**
5. Laporan tugas besar dikumpulkan pada melalui tautan berikut <https://bit.ly/SubmitTubes2Algeo> dengan link repository pada laporan. (Dibuat public setelah deadline pengumpulan).
6. Jika terdapat revisi yang menyebabkan pengumpulan harus dilakukan lebih dari sekali, silahkan kumpulkan dalam berkas dengan format penamaan **Algeo02-XXXXX-RevisiX** dengan X adalah nomor revisi.
7. Deadline pengumpulan adalah **Senin, 22 November 2022** pukul 23.59 WIB. Pengumpulan setelah waktu tersebut akan mendapat pengurangan nilai.

PENILAIAN

Komposisi penilaian umum adalah sebagai berikut :

1. Program: 80 %
2. Laporan: 20 %

REFERENSI

Perhatikan bahwa referensi hanya ada sebagai sumber belajar, bukan sebagai standar apa yang kami ekspektasi dari kode kalian. Banyak sumber lain, jangan menutup diri ke hanya sumber sumber di bawah ini.

“Nilai Eigen dan Vektor Eigen Bagian 1” by Rinaldi Munir

[https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18-
Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18-
Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf)

“Face Recognition Using Eigenfaces (PCA Algorithm)

<https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/>

“Pengenalan Wajah Menggunakan Algoritma Eigenface dan Euclidean Distance

[https://www.neliti.com/publications/135138/pengenalan-wajah-menggunakan-algoritma-
eigenface-dan-euclidean-distance](https://www.neliti.com/publications/135138/pengenalan-wajah-menggunakan-algoritma-
eigenface-dan-euclidean-distance)

“Python GUI Programming With Tkinter”

<https://realpython.com/python-gui-tkinter/>

“Create UI in Python-Tkinter”

<https://www.tutorialsteacher.com/python/create-gui-using-tkinter-python>

“How to build a GUI with PyQt”

<https://blog.logrocket.com/how-to-build-gui-pyqt/>

“Feature extraction and similar image search with OpenCV for newbies”

[https://medium.com/machine-learning-world/feature-extraction-and-similar-image-
search-with-opencv-for-newbies-3c59796bf774](https://medium.com/machine-learning-world/feature-extraction-and-similar-image-
search-with-opencv-for-newbies-3c59796bf774)

Selamat Mengerjakan!

“Ithamsanqa kunye nethemba lokuhle”

- Kyle -

“”

- Ziyad -

Believe in yourself :)

- Angel -

“https://youtu.be/k49w4_ZfaWM”

- Vincent -

“Belum waktunya keluar ITB :)”

- Vionie -

“<https://bit.ly/QuotesDariSaul>”

- Saul -