

**TUGAS BESAR 2 IF2123
Aljabar Linear dan Geometri
2022/2023**



KELOMPOK 40 - DigOldBicks

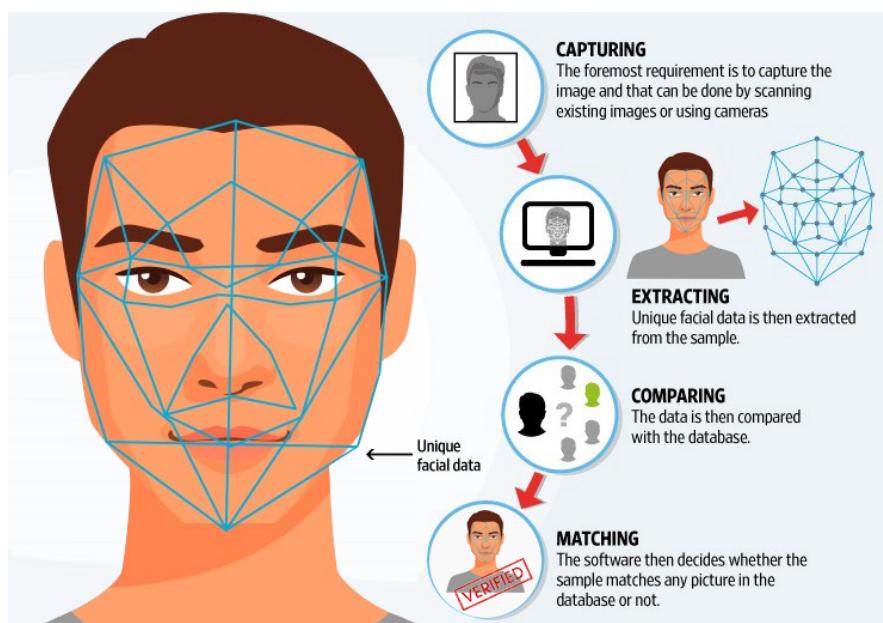
Frankie Huang	13521092
Johann Christian Kandani	13521138
Irgiansyah Mondo	13521167

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

BAB 1

DESKRIPSI MASALAH

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 pencocokan. 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.

BAB 2

TEORI SINGKAT

2.1 Perkalian Matriks

Perkalian matriks adalah salah satu jenis operasi matriks yang paling sering digunakan. Perkalian matriks hanya dapat dilakukan dengan dua buah matriks, dimana ukuran kolom matriks pertama harus sama dengan ukuran baris matriks kedua. Perkalian matriks bersifat tidak komutatif, sehingga perkalian matriks $AB \neq BA$. Secara singkat, rumus perkalian matriks dapat dinyatakan sebagai berikut

$$\begin{aligned} C &= AB \\ C_{ik} &= \sum_{j=0}^m A_{ij} B_{jk} \end{aligned}$$

Dimana A , B , dan C adalah matriks; ik adalah pasangan koordinat matriks, dan m adalah ukuran kolom matriks pertama atau ukuran baris matriks kedua.

2.2 Nilai Eigen dan Vektor Eigen

Jika terdapat sebuah matriks A dengan ukuran $n \times n$, maka vektor tidak-nol x di R^n disebut vektor eigen dari A jika Ax sama dengan perkalian skalar λ dengan x , yaitu

$$Ax = \lambda x$$

Skalar λ disebut *nilai eigen* dari A , dan x dinamakan vektor eigen yang berkoresponden dengan λ . Dengan kata lain, operasi $A\lambda$ dan Ax akan menghasilkan suatu vektor yang sama, yaitu vektor x yang akan menyusut atau memanjang dengan faktor λ .

Nilai eigen dan vektor eigen dapat dihitung dengan rumus sebagai berikut

$$\begin{aligned} Ax &= \lambda x \\ IAx &= \lambda Ix \\ Ax &= \lambda Ix \\ (\lambda I - A)x &= 0 \end{aligned}$$

$x = 0$ adalah solusi trivial dari persamaan di atas. Agar memiliki solusi tidak nol, maka

$$\det(\lambda I - A) = 0$$

Persamaan di atas disebut dengan persamaan karakteristik dari matriks A dan akar-akar persamaan tersebut, yaitu λ , dinamakan akar-akar karakteristik atau nilai-nilai eigen.

2.3 Eigenface

Jika A adalah matriks wajah-wajah training yang telah dinormalkan, maka matriks kovarian dari matriks wajah tersebut dapat dinyatakan sebagai

$$C = AA^T$$

Namun, karena matriks A berdimensi $N^2 \times M$, mengkomputasikan vektor-vektor eigen membutuhkan proses yang banyak, sehingga digunakan matriks kovarian

$$C' = A^T A$$

Vektor-vektor eigen dari C dapat dinyatakan dalam

$$C'v_i = A^T A v_i$$

$$\lambda_i v_i = A^T A v_i$$

$$A\lambda_i v_i = AA^T A v_i$$

$$\lambda_i A v_i = C A v_i$$

$$\lambda_i u_i = C u_i$$

Vektor-vektor eigen u_i adalah *EigenFace* dari matriks wajah A . *EigenFaces* tersebut dijadikan vektor-vektor basis dari vektor-vektor wajah yang telah dinormalkan

BAB 3

IMPLEMENTASI PROGRAM

3.1. Tech Stack/Kakas yang Digunakan

Program ini menggunakan beberapa kakas dari library python di antaranya:

Library	Method	Deskripsi
numpy	numpy.array	Konversi sebuah data menjadi tipe <i>ndarray</i>
	numpy.dot	Digunakan dalam mengkomputasi dot product dua buah vektor
	numpy.eye	Digunakan dalam menghasilkan

		matriks identitas berdimensi n*n
	numpy.mean	Digunakan dalam mengkomputasi rata-rata kumpulan vektor menjadi sebuah vektor rata-rata
	numpy.transpose	Digunakan dalam operasi transpos sebuah matriks
	numpy.polynomial.polynomial.polyadd	(tidak digunakan dalam versi saat ini) Digunakan dalam menjumlahkan dua buah polinom
	numpy.polynomial.polynomial.polymul	(tidak digunakan dalam versi saat ini) Digunakan dalam mengalikan dua buah polinom
	numpy.polynomial.polynomial.polyroots	(tidak digunakan dalam versi saat ini) Digunakan dalam mengkomputasikan akar-akar dari sebuah polinom
cv2	cv2.cvtColor	Mengubah <i>colorspace</i> sebuah gambar
	cv2.COLOR_BGR2RGB	Parameter dari cv2.cvtColor, digunakan untuk mengubah gambar menjadi grayscale
	cv2.VideoCapture	Digunakan untuk menangkap gambar melalui kamera
time	time.time	Digunakan untuk mencatat waktu pada saat method dijalankan
os	os.path.exists	Mengecek apakah sebuah folder sudah ada
	os.makedirs	Membuat sebuah direktori
	os.listdir	Mencetak semua file dan folder yang ada pada suatu direktori

PIL	PIL.Image.open	Membuka sebuah objek Image melalui path
	PIL.Image.resize	Mengubah dimensi sebuah objek Image
	PIL.Image.BICUBIC	parameter dari PIL.Image.resize, digunakan untuk memperbesar image dengan <i>bicubic interpolation</i>
	PIL.Image.fromarray	Membuat sebuah image dari sebuah array
	PIL.Image.crop	Memotong sebuah image dengan dimensi yang telah dispesifikasi
	PIL.Image.convert	Mengubah <i>color space</i> sebuah image
	PIL.ImageTk.PhotoImage	Digunakan untuk membaca file ekstensi .jpg
tkinter	tkinter.Tk	Digunakan untuk membuat sebuah window
	tkinter.Canvas	Digunakan untuk membuat sebuah objek Canvas untuk meletakkan gambar
	tkinter.Entry	Digunakan untuk membaca sebuah string dari user
	tkinter.Text	Digunakan untuk membuat sebuah objek Text untuk menampilkan tulisan
	tkinter.Button	Digunakan untuk membuat sebuah objek Button
	tkinter.PhotoImage	Digunakan untuk membaca file ekstensi .png
	tkinter.filedialog	Digunakan untuk membuka sebuah file/direktori

3.2. Algoritma Program

Program ini terdiri dari 3 file utama, yaitu:

File	Function	Deskripsi
eigenfunction.py	polyKofaktor(M)	(Sudah tidak digunakan) Mengkomputasikan polinom persamaan karakteristik eigen value matriks persegi
	eigenValue1(A)	Mengkomputasikan nilai eigen dengan metode persamaan karakteristik eigen value dari sebuah matriks persegi
	eigenVectors1(A, eigVal)	Mengkomputasikan vektor eigen sebuah matriks berdasarkan sebuah nilai eigen matriks tersebut
	QRsqHouseholder(A)	Mengkomputasikan dekomposisi QR sebuah matriks persegi dengan algoritma householder.
	eigen(A, iteration=10000)	Mengkomputasikan nilai-nilai eigen dari sebuah matriks persegi (beserta vektor-vektor eigen jika matriks simetris) dengan algoritma QR. Default program akan menjalankan 10000 iterasi QR yang diperkirakan dapat menghampiri nilai eigen dan vektor eigen dengan cukup baik.
image_processing.py	train_images(path)	Membaca direktori berisi training images dan mengembalikan array berisi direktori file gambar, nilai muka rata-rata, nilai eigenface, dan nilai omega

	<code>test_image(img, pict_name, mean_face, EigFace, 0m)</code>	Mengembalikan direktori file gambar dengan tingkat error paling kecil
main.py	<code>load_data()</code>	Membaca nilai-nilai yang disimpan pada direktori data/
	<code>relative_to_assets (path)</code>	Mengembalikan string berisi direktori assets/
	<code>select_dir()</code>	Membuka file browser dan memilih direktori yang akan digunakan sebagai direktori training images
	<code>change_image(img, path)</code>	Mengubah gambar pada kotak Test Image dan Closest Result
	<code>select_file()</code>	Membuka file browser dan memilih gambar yang akan digunakan sebagai test image
	<code>run_camera()</code>	Menyalakan kamera dan menggunakan sebagai test image
	<code>change_mode()</code>	Mengganti antara mode kamera dan mode input image

Image Pre-processing

Citra diubah menjadi citra grayscale berukuran 256 x 256 pixel, citra-citra tersebut kemudian diubah dalam bentuk matriks yang diubah menjadi sebuah vektor-vektor $256^2 \times 1$. Vektor-vektor citra di "normal"kan dengan cara dikurangi dengan rata-rata vektor-vektor tersebut, kemudian disatukan dalam sebuah matriks sebagai kolom-kolom matriks.

Kemudian dengan matriks wajah-wajah tersebut, hitung matriks kovarian yang dicari nilai-nilai eigen dan vektor-vektor eigen matriks kovarian. Matriks wajah kemudian dikalikan dengan vektor-vektor eigen untuk menghasilkan vektor-vektor eigenface. Vektor-vektor wajah kemudian diubah

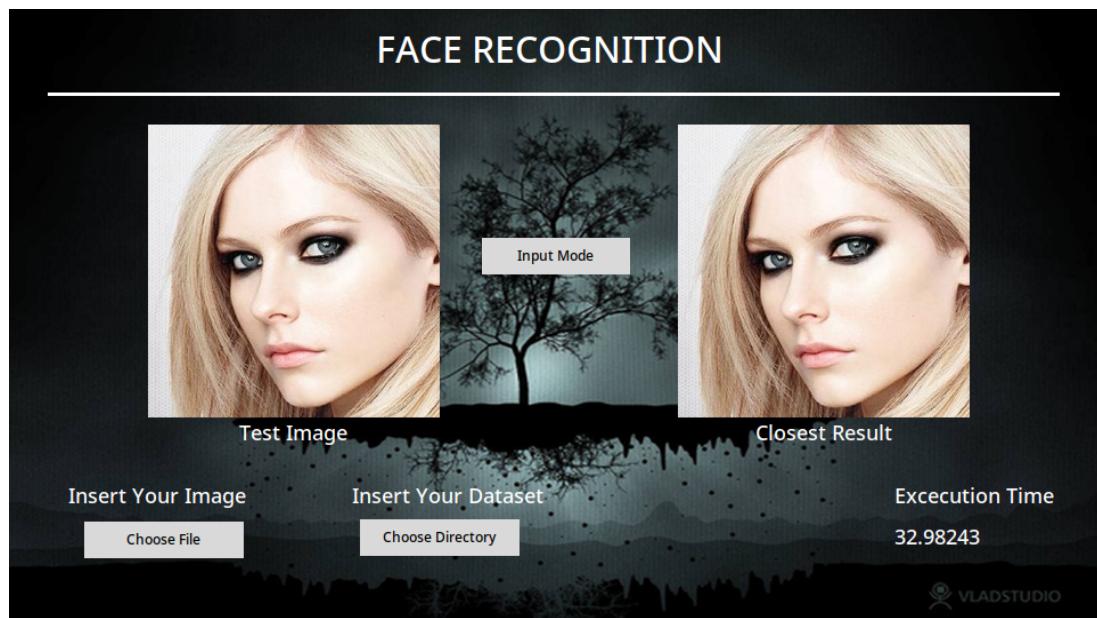
dalam bentuk vektor konstanta-konstanta kombinasi linear dari vektor-vektor eigenface tersebut.

Image Matching

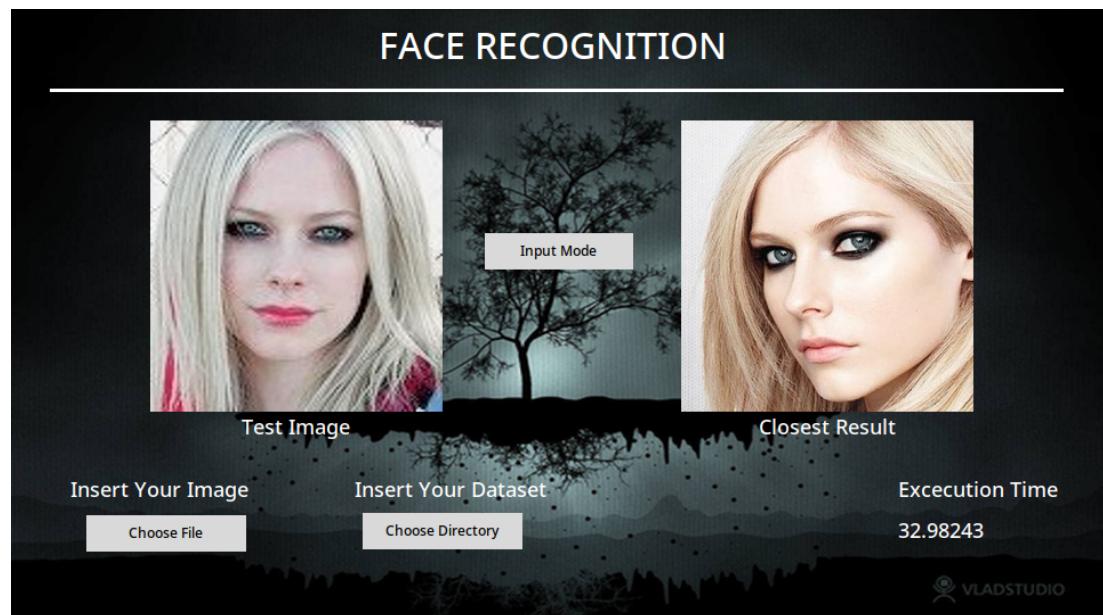
Citra wajah diproses seperti langkah pre-processing menjadi vektor wajah yang telah di"normal"kan dengan wajah rata-rata. Vektor wajah tersebut kemudian dinyatakan juga dalam kombinasi linear vektor-vektor eigenface. Vektor tersebut kemudian dihitung jarak euclidean dengan vektor-vektor lain, dan pilih wajah dengan jarak euclidean terkecil.

BAB 4 EKSPERIMEN

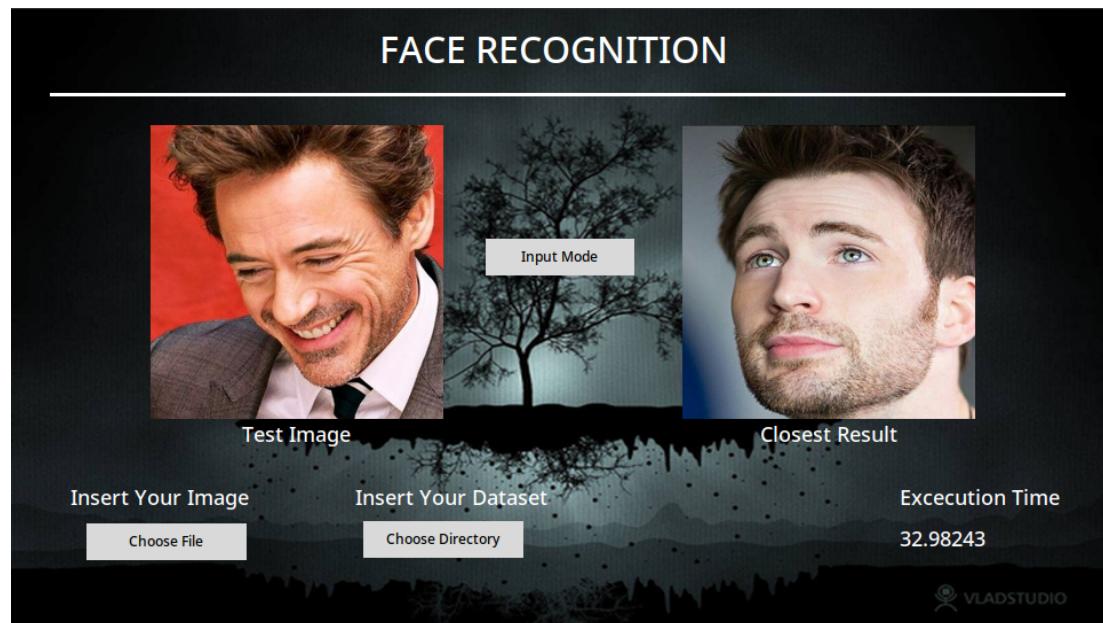
4.1. Gambar terdapat dalam dataset



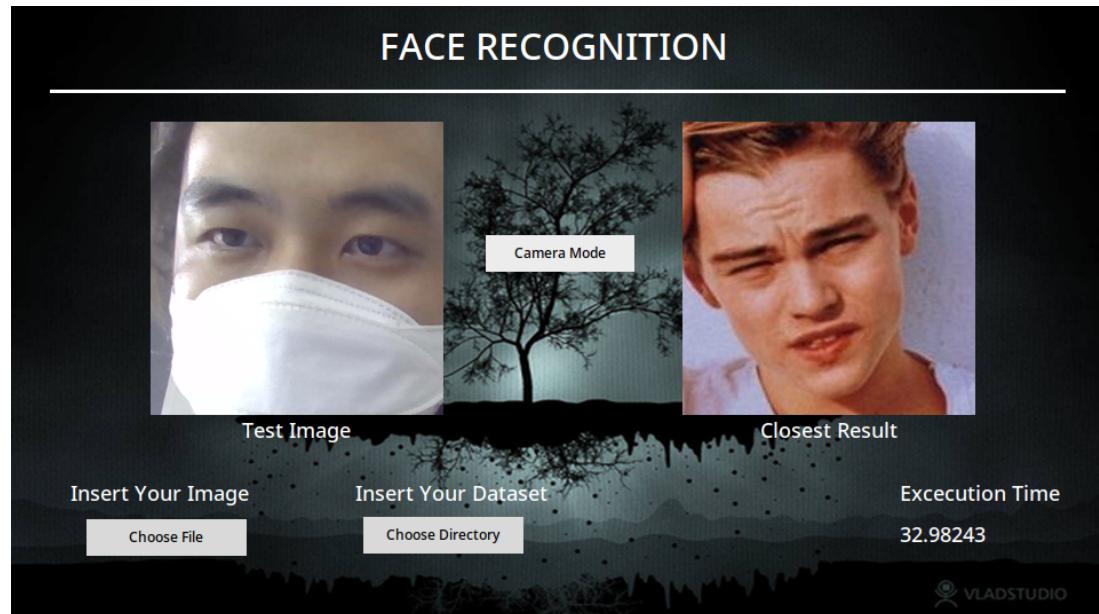
4.2. Gambar tidak terdapat dalam dataset



4.3. Wajah tidak terdapat dalam dataset



4.4. Wajah diambil menggunakan kamera



BAB 5 KESIMPULAN

Pada tugas kali ini, kami berhasil membuat program GUI yang dapat melakukan *face recognition* yang menerima file gambar dan dataset sehingga program tersebut dapat mengenali dari objek yang sesuai dengan permintaan. Selain itu, kami juga banyak belajar manajemen waktu dan manajemen kelompok yang memberikan pembelajaran yang berharga buat kami untuk saling bekerjasama di tugas-tugas yang lain di jurusan Informatika ini.

DAFTAR REFERENSI

- <https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/>
- <https://numpy.org/doc/stable/reference/>
- <https://www.kaggle.com/datasets/herveisburak/pins-face-recognition>
- <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18-Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf>
- <https://github.com/ParthJadhav/Tkinter-Designer>

LAMPIRAN

Github: <https://github.com/frankiehuangg/Algeo02-21092>