

Laporan Tugas Besar 2
IF 2123 Aljabar Linier dan Geometri
“Aplikasi Nilai Eigen dan EigenFace pada Pengenalan Wajah
(*Face Recognition*)”



Disusun oleh:
Kelompok 5 (IndomieLovers)

Anggota:

- Manuella Ivana Uli Sianipar (13521051)
- Alex Sander (13521061)
- Moh. Aghna Maysan Abyan (13521076)

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

SEMESTER I TAHUN AJARAN 2022/2023

DAFTAR ISI

	Hal
DAFTAR ISI	1
BAB I Deskripsi Masalah	2
BAB II Teori Singkat	3
2.1 Perkalian Matriks	3
2.2 Nilai Eigen	3
2.3 Vektor Eigen	4
2.4 Eigenface	4
2.5 Euclidian Distance	5
BAB III Implementasi Pustaka dan Program Dalam Python	6
BAB IV Eksperimen	7
BAB V Kesimpulan, Saran, dan Refleksi	10
5.1 Kesimpulan	10
5.2 Saran	10
5.3 Refleksi	10
REFERENSI	11
REPOSITORY GITHUB	11

BAB I

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.

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.

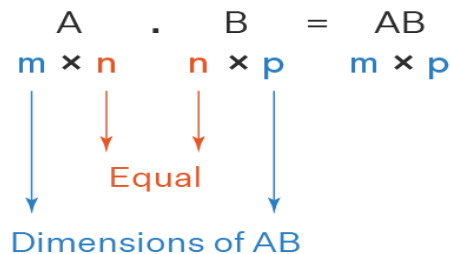
Program dibuat dengan Bahasa Python dengan memanfaatkan sejumlah library di OpenCV (Computer Vision) atau library pemrosesan gambar lainnya (contoh PIL). Fungsi untuk mengekstraksi fitur dari sebuah citra wajah tidak perlu anda buat lagi, tetapi menggunakan fungsi ekstraksi yang sudah tersedia di dalam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

BAB II

TEORI SINGKAT

2.1 Perkalian Matriks

Peraturan perkalian matriks menyatakan bahwa 2 buah matriks, misalkan matriks A dan B, dapat menghasilkan perkalian apabila jumlah kolom matriks A = jumlah baris matriks B. Apabila jumlah kolom matriks A \neq jumlah baris matriks B, perkalian matriks A dan matriks B tidak terdefinisi.



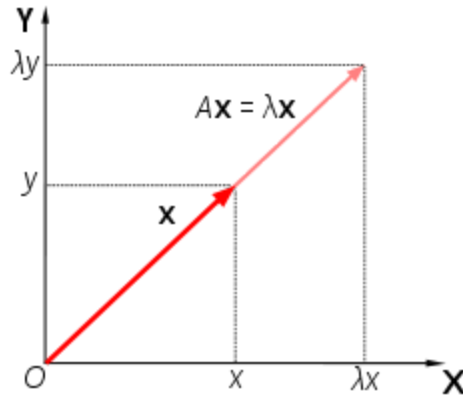
Misalkan, terdapat matriks A berukuran 2×3 dan matriks B berukuran 3×4 , maka hasil perkalian matriks adalah matriks AB dengan ukuran 2×4 . Contoh lainnya adalah jika matriks A memiliki ukuran 2×3 dan matriks B memiliki ukuran 2×3 , perkalian matriksnya tidak terdefinisi dikarenakan jumlah kolom matriks A \neq jumlah baris matriks B ($3 \neq 2$).

2.2 Nilai Eigen

Nilai eigen (*Eigenvalue*) menyatakan nilai karakteristik dari sebuah matriks berukuran $n \times n$. Rumus dasar dari nilai eigen adalah:

$$Ax = \lambda x$$

Dengan λ bersifat sebagai nilai eigen dari matriks A berukuran $n \times n$. x pada rumus diatas dinamakan sebagai vektor eigen yang berkoresponden dengan nilai eigen. Vektor eigen akan dibahas lebih lanjut pada subbab selanjutnya. Nilai eigen berpengaruh pada vektor eigen, karena nilai eigen dapat membuat vektor berbentuk lebih panjang atau lebih pendek dari bentuk awal. Untuk lebih jelasnya, dapat dilihat gambar dibawah ini.



2.3 Vektor Eigen

Sesuai penjelasan pada subbab Nilai Eigen, vektor eigen adalah vektor kolom yang direpresentasikan dengan simbol x , dimana jika vektor eigen dikalikan dengan matriks $n \times n$ akan menghasilkan vektor lain yang merupakan kelipatan dari vektor eigen.

Vektor eigen dapat dicari melalui rumus:

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

Dengan I adalah matriks identitas. Nilai karakteristiknya dapat dicari melalui rumus:

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

2.4 Eigenface

Eigenface adalah istilah yang diberikan untuk sekumpulan vektor eigen pada permasalahan *computer vision* (penglihatan komputer) pada pengenalan wajah manusia. Algoritma eigenface dapat digunakan untuk basis face recognition sederhana.

Langkah-langkah dalam proses algoritma eigenface adalah sebagai berikut.

1. Menyiapkan data ke dalam suatu himpunan S yang berisi semua training image yang disusun menjadi matrix $n^2 \times 1$.

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

2. Mencari nilai tengah atau mean dari semua training images tersebut.

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

3. Mencari selisih antara nilai training image dan mean.

$$\phi_i = \Gamma_i - \Psi$$

4. Menghitung matriks kovarian

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T$$

5. Menghitung eigenvalue dan eigen vektor dari matriks kovarian

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

6. Mencari eigenface dari eigenvalue dan eigenvektor yang diperoleh

$$\mu_i = \sum_{k=1}^M v_{ik} \phi_k$$

2.5 Euclidian Distance

Euclidian distance method adalah salah satu metode untuk mencari jarak terpendek. Metode ini digunakan untuk mencari distance dari nilai eigen training images dalam database dengan nilai eigen dari image testface.

$$\varepsilon_k = \Omega - \Omega_k$$

Method

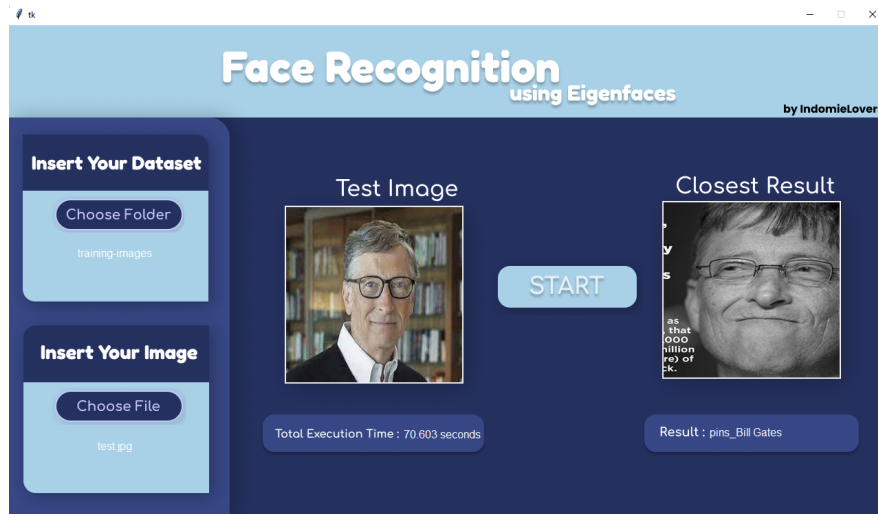
IF2123 - 6

BAB IV EKSPERIMEN

Percobaan 1:

Jumlah Training Images : 1002

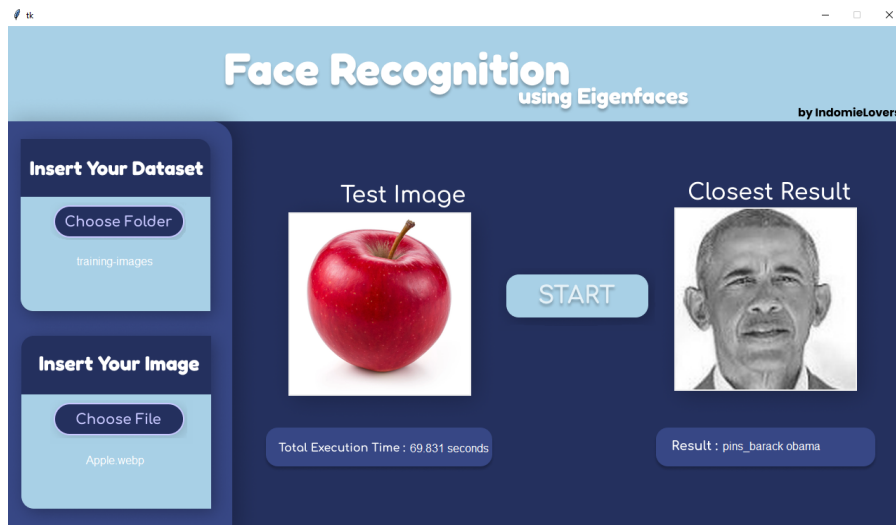
Jumlah Set(Nama): 7



Percobaan 2:

Jumlah Training Images : 1002

Jumlah Set(Nama): 7

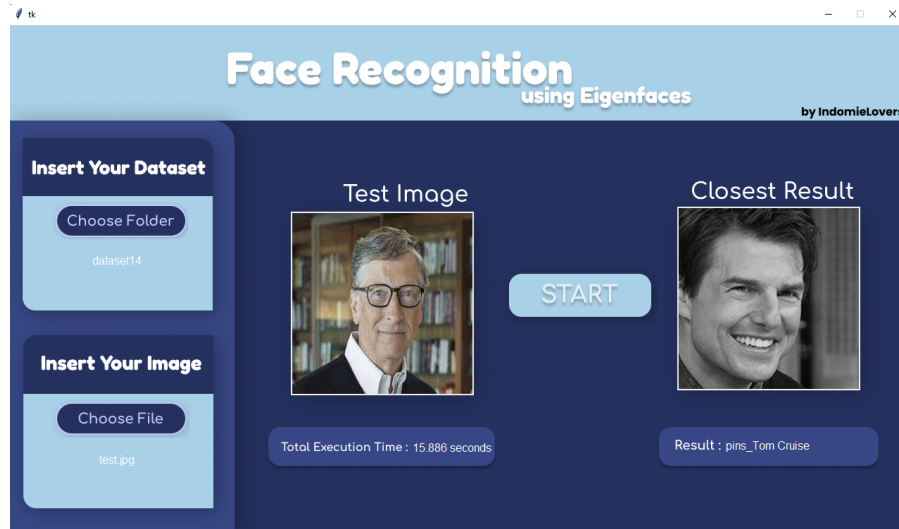


Kesalahan yang muncul dapat diakibatkan karena *threshold* yang digunakan adalah setengah dari jarak terbesar dari seluruh training images. Diduga kesalahan muncul karena *threshold* yang didapatkan jauh lebih besar dari dugaan.

Percobaan 3:

Jumlah Training Images: 379

Jumlah Set(Nama): 27

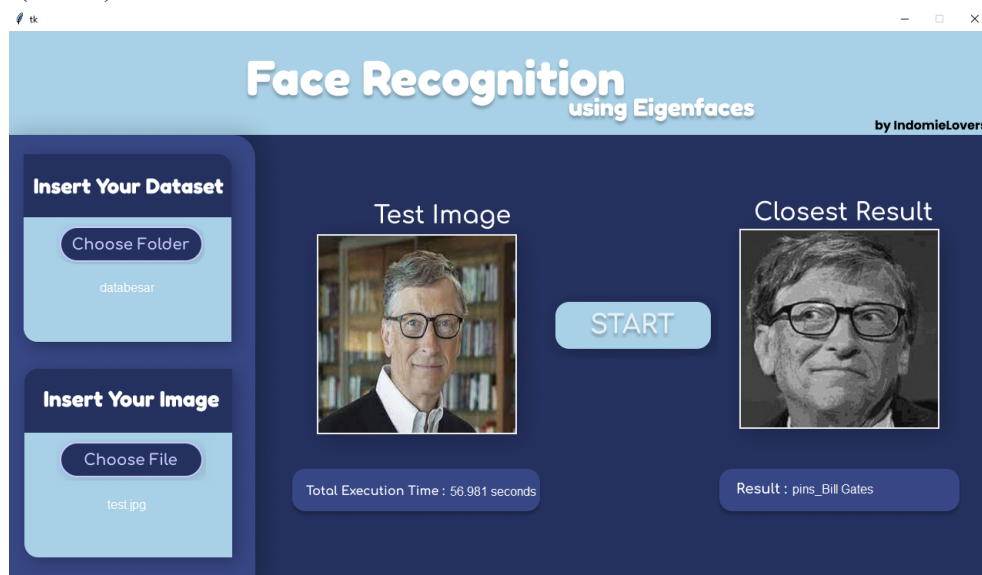


Kesalahan yang muncul diduga karena sedikitnya training-images yang digunakan.

Percobaan 4:

Jumlah Training Images: 882

Jumlah Set(Nama): 10

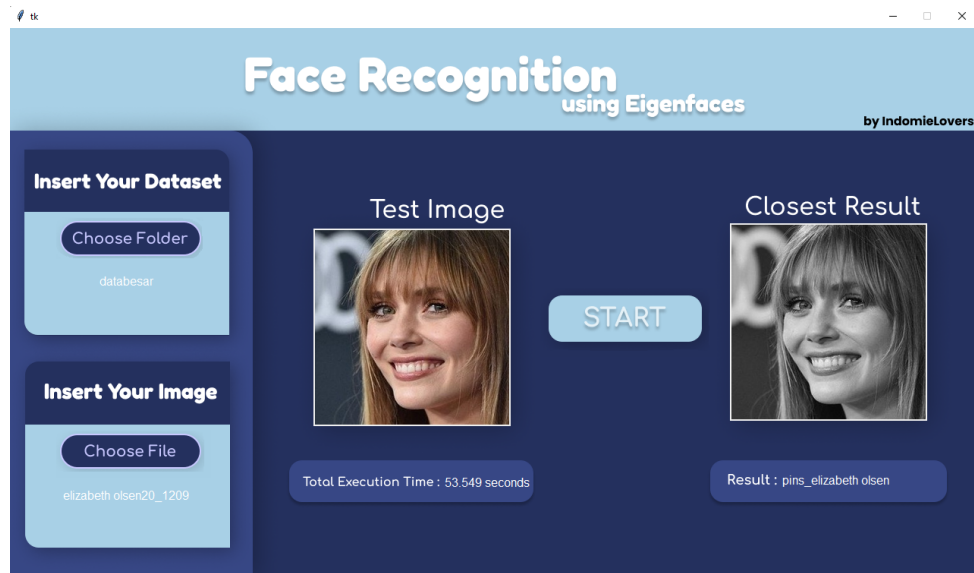


Percobaan 5:

Jumlah Training Images: 882

Jumlah Set(Nama): 10

Note : Penggunaan foto yang sama dari dataset



BAB V

KESIMPULAN, SARAN, DAN REFLEKSI

5.1 Kesimpulan

Ilmu mengenai nilai eigen dan vektor eigen dapat diaplikasikan melalui banyak hal dalam kehidupan nyata, salah satunya adalah untuk pengenalan wajah seperti yang telah dijelaskan pada tugas besar kali ini. Untuk melakukan hal-hal tersebut, diperlukan pengetahuan dasar mengenai konsep nilai eigen dan vektor eigen serta metode-metode yang diperlukan untuk mendapatkannya. Konsep ini dapat dilakukan dengan efisien, namun konsep ini memiliki beberapa kekurangan, seperti muka subjek harus tepat menghadapi kamera, algoritma yang sensitif terhadap pencahayaan, dan diperlukannya muka subjek tepat di tengah foto. Oleh karena itu, konsep ini jarang digunakan. Namun, ini dapat digunakan untuk skala kecil, seperti face recognition lokal untuk gawai pribadi.

5.2 Saran

Untuk saran kedepannya, kami berharap tugas besar dapat disertakan dengan asistensi untuk mempermudah mahasiswa dalam mengerjakan dan memastikan apakah hal yang sudah dikerjakan berjalan dengan baik atau tidak. Selain itu, kami juga berharap referensi untuk file panduan tugas besar ditambahkan agar memperjelas proses pengerjaan.

5.3 Refleksi

Pada tugas besar kedua Aljabar Linier dan Geometri semester ini, kami merasa masih kurang cekatan dan kurang taktis dalam mengerjakan tugas besar ini sehingga tidak sempat mengerjakan bonus yang diberikan. Tetapi, kami tetap mengapresiasi teman-teman sekelompok yang tidak menghilang ketika diajak mengerjakan tugas besar.

REFERENSI

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

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

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

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

REPOSITORY GITHUB

<https://github.com/manuellaiv/Algeo02-21051>