

LAPORAN TUGAS BESAR 2 IF2123 2022
APLIKASI NILAI EIGEN DAN VEKTOR EIGEN DALAM KOMPRESI GAMBAR

Disusun untuk memenuhi salah satu tugas besar mata kuliah IF2123 Aljabar Linear Geometri
pada Semester I Tahun Akademik 2022/2023

Disusun oleh:

Hidayatullah Wildhan Ghaly B.	13521015
Ditra Rizqa Amadia	13521019
Fahrian Afdholi	13521031



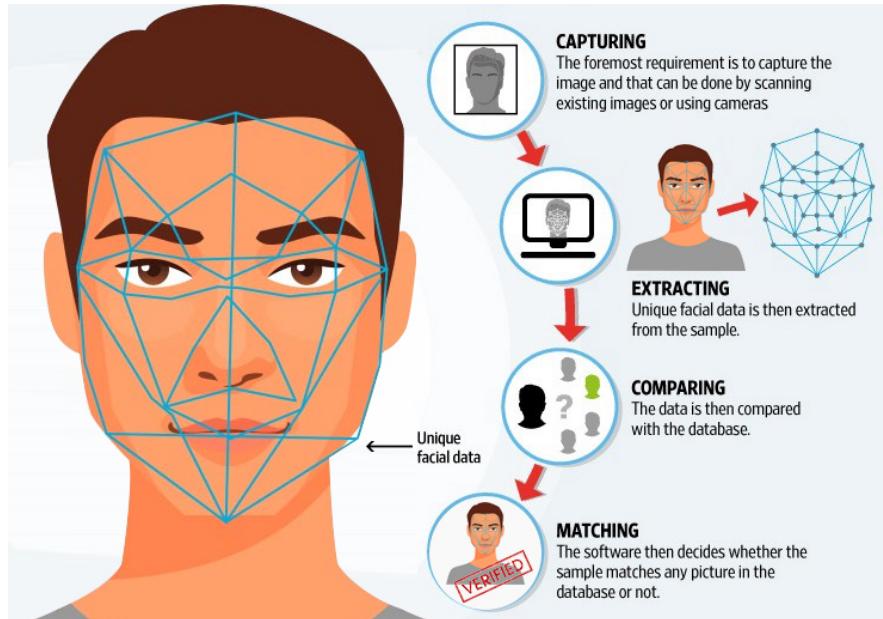
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2022

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. 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.

BAB II

TEORI SINGKAT

Eigenface adalah salah satu algoritma pengenalan wajah yang berdasarkan pada Principle Component Analysis (PCA) yang dikembangkan di MIT. Algoritma EigenFace secara keseluruhan cukup sederhana, Training Image direpresentasikan dalam sebuah vektor flat (gabungan vektor) dan digabung bersama-sama menjadi sebuah matriks tunggal. Eigen Vector kemudian diekstraksi dan disimpan dalam file temporary atau database. Training image kemudian diproyeksikan dalam feature space yang dinamai face space yang ditentukan oleh eigen vektor(Mukti, 2008). Berikut adalah langkah pengerjaan 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 rata-rata 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)

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

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

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.

BAB III

IMPLEMENTASI PROGRAM

3.1 Penggerjaan

Salah satu library yang digunakan untuk penggerjaan program ini adalah library OpenCV. OpenCV adalah open source library untuk image processing dan computer vision. Secara teori OpenCV digunakan seperti meniru cara kerja sistem visual manusia yaitu dengan melihat objek melalui "penglihatan/mata" dan citra pada objek tersebut diteruskan ke otak untuk memproses sehingga mengerti objek apa yang tampak pada pandangan mata manusia. OpenCV merupakan salah satu cabang Artificial intelligence (kecerdasan buatan) yang digunakan untuk pengembangan atau analisis isi suatu gambar.

Selain itu, pada program ini juga digunakan library numpy. NumPy (Numerical Python) adalah library python yang digunakan untuk bekerja dengan array dan juga memiliki fungsi yang bekerja dalam domain aljabar linier, transformasi fourier, dan matriks. Library yang dibuat pada 2005 oleh Travis Oliphant ini merupakan proyek open source sehingga Anda dapat menggunakan secara bebas. Meski python memiliki daftar yang melayani tujuan array, prosesnya begitu lambat sehingga memerlukan NumPy yang bisa menyediakan objek array hingga 50 kali lebih cepat daripada daftar python tradisional.

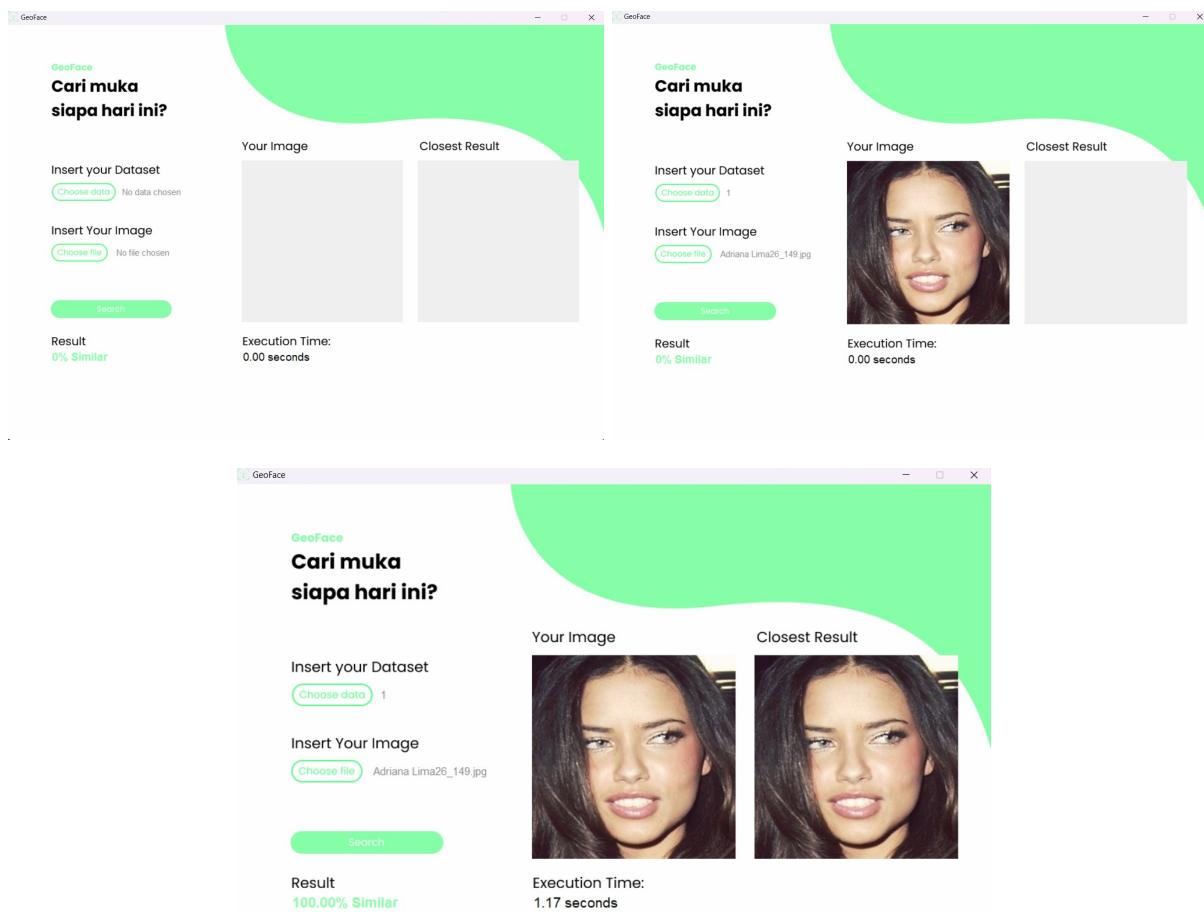
Program ini menggunakan perhitungan seperti eigen face, eigen value, eigen vector, covarian, dan nilai rata-rata dari si matrix. Dalam perhitungan eigen face program ini menggunakan eigen vektor dan nilai rata rata dari matrix gambar. Sebelum mendapatkan eigen face program ini diharuskan untuk membuat eigen vektor dan eigen value terlebih dahulu dilakukan dengan cara mengalikan lamda dengan identitas dan menguranginya dengan matriks, setelah itu cari valuenya dengan mendeterminankan matriksnya dan mencari value dari si lamdadand setelah mendapatkannya eigen vektor bisa di dapatkan. Covaria sendiri dicari dengan mengalikan si matrix dengan transpose dirinya sendiri, sedangkan nilai rata-rata matriks dicari dengan menjumlahkan seluruh matriksnya menjadi satu matriks lalu membaginya sesuai dengan jumlah matriks yang ada sebelumnya.

BAB IV

EKSPERIMEN

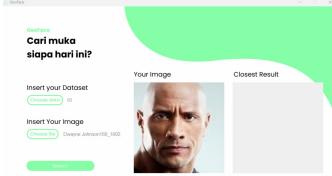
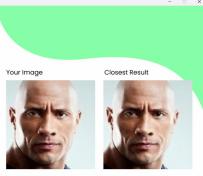
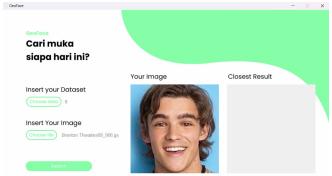
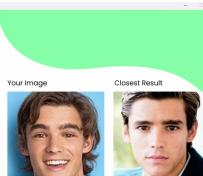
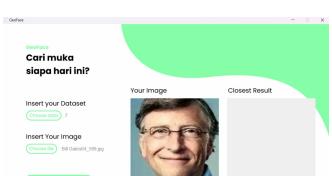
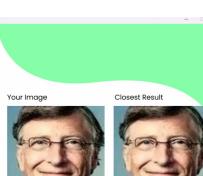
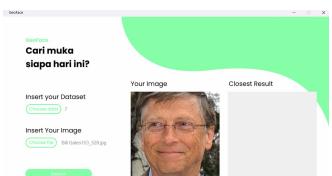
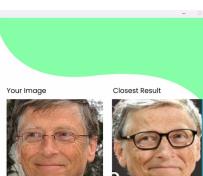
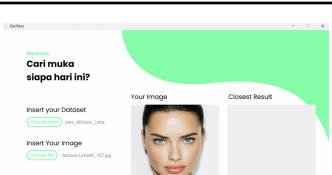
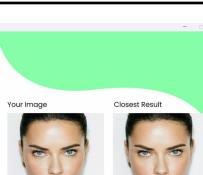
4.1 Tampilan Dalam Program

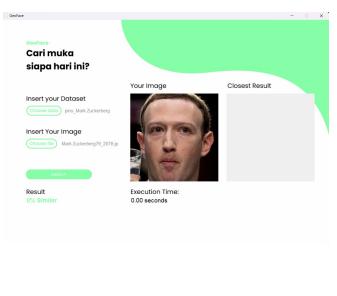
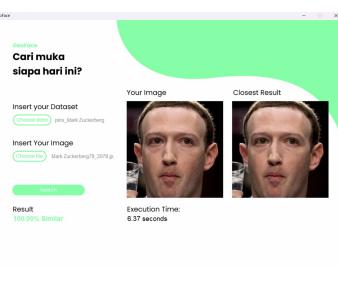
Berikut tampilan program saat dijalankan:



4.2 Hasil Face Recognition

Pada eksperimen yang kami lakukan, kami melakukan uji coba beberapa gambar dalam bentuk jpg untuk dilakukan face recognition. Uji coba yang telah kami lakukan adalah sebagai berikut:

No	Waktu eksekusi	Gambar awal	Gambar akhir	Deskripsi
1	0.97 detik			Ukuran File: 14.1 KB Ukuran Gambar: 209 x 221 File type: JPG
2	0.97 detik			Ukuran File: 11.8 KB Ukuran Gambar: 145 x 155 File type: JPG
3	0.95 detik			Ukuran File: 8.85 KB Ukuran Gambar: 121 x 129 File type: JPG
4	0.93 detik			Ukuran File: 18.0 KB Ukuran Gambar: 194 x 221 File type: JPG
5	15.58 detik			Ukuran File: 44.1 KB Ukuran Gambar: 329 x 387 File type: JPG

6	6.37 detik			Ukuran File: 33.9 KB Ukuran Gambar: 302 x 320 File type: JPG
---	---------------	-----------------------------------------------------------------------------------	------------------------------------------------------------------------------------	--------------------------------------------------------------------

Pada percobaan nomor pertama dan ketiga, kami menggunakan dataset yang memiliki lima buah foto. Input dari gambar sudah ada di dataset sehingga program akan membaca dan mengeluarkan output yang sama dengan input pada gambar. Hal ini dikarenakan program akan mendapatkan nilai euclidean distance bernilai nol. Waktu eksekusi dari kedua percobaan ini sangat sebentar karena dataset yang digunakan hanyalah lima buah.

Pada percobaan kedua dan keempat, diuji coba data set yang tetap berisi lima gambar. Hanya saja, input yang diberikan adalah gambar yang tidak ada di dalam dataset tersebut sehingga program akan menampilkan gambar yang paling mirip yang ada di dalam dataset dengan inputan. Waktu eksekusi dari kedua percobaan ini mirip dengan percobaan sebelumnya karena hanya menggunakan lima buah dataset.

Pada percobaan kelima, digunakan dataset yang memiliki 213 gambar dan input yang digunakan adalah gambar yang ada pada dataset tersebut. Program menampilkan gambar yang sama karena gambar input ada di dalam dataset sehingga program membaca euclidean distance bernilai nol. Untuk waktu eksekusinya sendiri lebih lama dibandingkan yang hanya menggunakan lima buah gambar di dalam dataset. Eksekusi timenya adalah 15.58 detik karena program harus mengolah banyak data.

Pada percobaan keenam, digunakan dataset yang memiliki 95 gambar dan inputnya berupa gambar yang ada di dalam dataset tersebut. Sama halnya dengan percobaan kelima, pengolahan datanya lebih lama dibandingkan dengan percobaan yang hanya menggunakan lima buah dataset. Tetapi tentunya percobaan ini lebih sebentar dibandingkan percobaan kelima karena dataset yang digunakan lebih sedikit. Waktu eksekusinya sendiri adalah 6.37 detik.

BAB V
KESIMPULAN, SARAN, DAN REFLEKS

5.1 Kesimpulan

Pembelajaran tentang nilai eigen merupakan salah satu pembelajaran yang penting dan bermanfaat dalam kehidupan sehari-hari. Dengan mengetahui konsep dari eigen, manusia menjadi lebih mudah untuk berkarya dan menemukan hal baru. Salah satu pemanfaatan eigen ini adalah pada pengenalan wajah. Dengan diketahuinya proses pengenalan wajah secara teori, maka manusia dapat membuat suatu program yang bisa digunakan untuk melakukan pengenalan wajah secara real time. Melalui pengenalan wajah, manusia dapat dengan mudah mengimplementasikannya di kehidupan nyata. Contohnya terdapat pada face ID.

Salah satu tahapan pengenalan wajah sendiri adalah dengan menentukan nilai eigen. Eigen vektor, eigen value, eigen face, covarian, dan nilai tengah dapat digunakan dalam menggunakan untuk menguji kemiripan foto dengan cara membandingkan satu foto yang diambil dengan dataset yg tersimpan

5.2 Saran

Saran untuk kedepannya adalah lebih teliti dalam membuat vektor egien dan value eigen dan berhati hati dalam menggunakan library. Agar meminimalisasi kesalahan, sebaiknya membuat driver pada setiap fungsi. Selain itu, jangan lupa untuk memberikan komentar pada setiap fungsi agar program lebih mudah untuk dibaca.

5.3 Refleks

Dengan adanya tugas besar ini kami dapat membangun pemikiran kami lebih teliti dan kreatif lagi dan dapat mengingkatkan kerjasama dalam bekelompok. Dengan adanya referensi yang sangat banyak, kami lebih mudah untuk membuat program yang menurut kami sudah benar.

DAFTAR REFERENSI

- [1] [9 Library Python yang Populer di Kalangan Data Scientist - Algoritma](#)
- [2] [Feature extraction and similar image search with OpenCV for newbies | by Andrey Nikishaev | Machine Learning World | Medium](#)
- [3] [jiptummpp-gdl-rendradwiw-43950-2-bab1.pdf](#)
- [4] [ML | Face Recognition Using Eigenfaces \(PCA Algorithm\) - GeeksforGeeks](#)
- [5] [Pengenalan Wajah Menggunakan Algoritma Eigenface Dan Euclidean Distance - Neliti](#)
- [6] [Python GUI Programming With Tkinter – Real Python](#)

LAMPIRAN

Repository Source Code Program

<https://github.com/fchrgrgrib/Algeo02-21015>

Video Presentasi

<https://youtu.be/rnEVxqsWb6I>