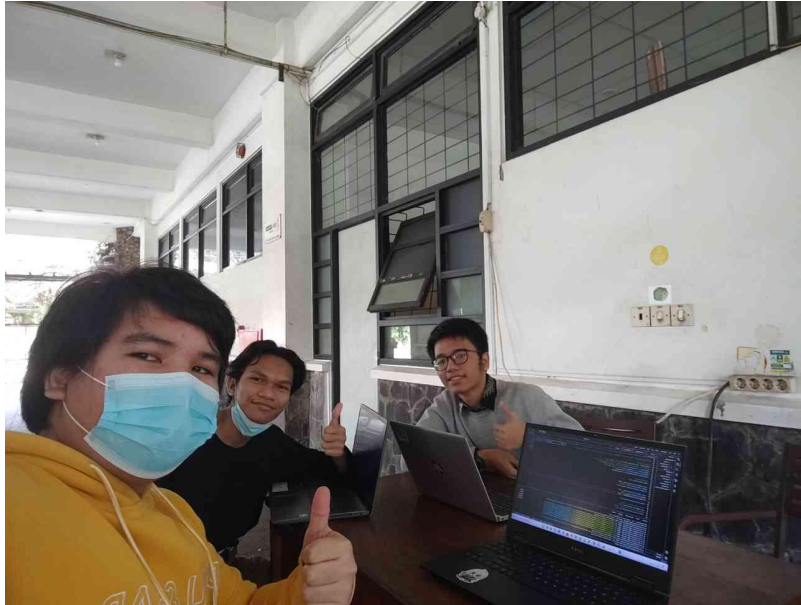


LAPORAN TUGAS BESAR 2
APLIKASI NILAI EIGEN DAN EIGENFACE PADA PENGENALAN
WAJAH (FACE RECOGNITION)

Aljabar Linier dan Geometri (IF2123)

Dosen Pengajar: Dr. Judhi Santoso, M.Sc.



Disusun Oleh:
Kelompok 48

Anggota Kelompok:
Kevin John Wesley Hutabarat - 13521042
Ezra Maringan Christian Mastra Hutagaol - 13521073
Moch Sofyan Firdaus - 13521083

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
NOVEMBER 2022

DAFTAR ISI

DAFTAR ISI.....	2
BAB 1 DESKRIPSI MASALAH.....	3
BAB 2 TEORI SINGKAT.....	4
1. Perkalian Matriks.....	4
2. Nilai Eigen dan Vektor Eigen.....	5
3. Eigenface.....	6
BAB 3 IMPLEMENTASI PROGRAM.....	8
1. Eigenvalue dan Eigenvector.....	8
2. Eigenface.....	8
3. Tahap pengenalan wajah.....	8
4. GUI.....	9
BAB 4 EKSPERIMEN.....	10
1. Eigenface.....	10
2. Tampilan Awal GUI.....	10
3. Pengenalan Wajah.....	11
BAB 5 KESIMPULAN, SARAN, REFLEKSI.....	13
1. Kesimpulan.....	13
2. Saran.....	13
3. Refleksi.....	13
DAFTAR REFERENSI.....	14
LAMPIRAN.....	15

BAB 1

DESKRIPSI MASALAH

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
7. Input gambar akan berukuran MINIMAL 256 X 256
8. Hasil training dapat disimpan, namun tetap wajib dapat dilakukan training kembali jika diperlukan user.

BAB 2

TEORI SINGKAT

1. Perkalian Matriks

Perkalian matriks adalah perkalian yang melibatkan suatu matriks atau susunan bilangan berupa kolom dan angka, serta memiliki sifat-sifat tertentu. Matriks adalah susunan bilangan, simbol, atau karakter yang disusun atas baris dan kolom seperti bangun persegi. Bilangan, simbol atau karakter di dalam matriks disebut dengan elemen dari matriks.

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad B = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

Matriks umumnya dinotasikan berupa huruf kapital seperti A dan B. Kemudian 1,2,3 dan 4 disebut elemen-elemen matriks A. Begitu pula a, b, c, d, e, f dan g elemen-elemen matriks B. Matriks memiliki ordo. Ordo merupakan bilangan yang menunjukkan banyak baris dan banyak kolom dari matriks. Ordo matriks A adalah 2×2 (jumlah baris 2 dan jumlah kolom 2). Dalam hal ini dapat ditulis

$$A_{2 \times 2}$$

Misalkan matriks A (a, b, c, d) berukuran 2×2 dikalikan dengan matriks B (e, f, g, h) berukuran 2×2 , sehingga rumusnya akan menjadi:

$$A \times B$$
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} j & k \\ l & m \end{pmatrix}$$
$$A \times B = \begin{pmatrix} aj + bl & ak + bm \\ cj + dl & ck + dm \end{pmatrix}$$

Syarat dua matriks dapat dioperasikan perkalian yaitu banyak kolom matriks pertama harus sama dengan banyak baris matriks kedua, sebagai berikut:

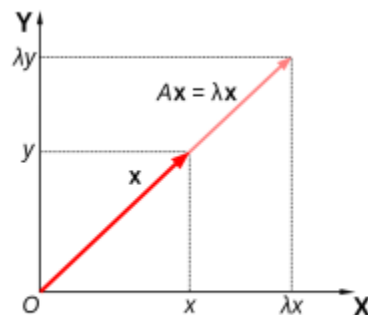
$$A_{m \times n} \times B_{n \times t} = C_{m \times t}$$

2. Nilai Eigen dan Vektor Eigen

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

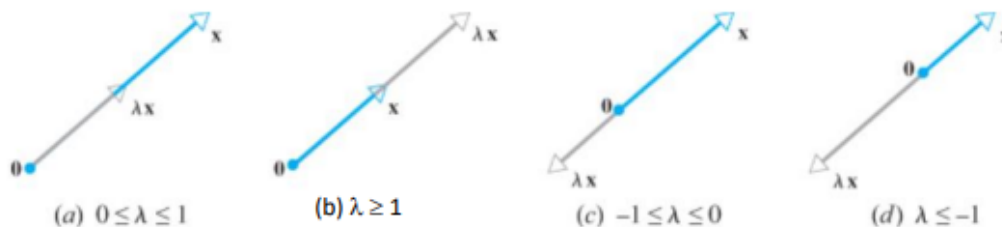
$$Ax = \lambda x$$

Skalar λ disebut nilai eigen dari A , dan x dinamakan vektor eigen yang berkoresponden dengan λ . Kata “eigen” berasal dari Bahasa Jerman yang artinya “asli” atau “karakteristik”. Dengan kata lain, nilai eigen menyatakan nilai karakteristik dari sebuah matriks yang berukuran $n \times n$. Vektor eigen x menyatakan matriks kolom yang apabila dikalikan dengan sebuah matriks $n \times n$ menghasilkan vektor lain yang merupakan kelipatan vektor itu sendiri.



Sumber gambar: Wikipedia

Dengan kata lain, operasi $Ax = \lambda x$ menyebabkan vektor x menyusut atau memanjang dengan faktor λ dengan arah yang sama jika λ positif dan arah berkebalikan jika λ negatif.



Berikut cara menghitung nilai eigen dan vektor eigen :

Diberikan sebuah matriks A berukuran $n \times n$. Vektor eigen dan nilai eigen dari matriks A dihitung sebagai berikut:

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

$\mathbf{x} = 0$ adalah solusi trivial dari $(\lambda I - A)\mathbf{x} = 0$. Agar $(\lambda I - A)\mathbf{x} = 0$ memiliki solusi tidak-nol, maka haruslah $\det(\lambda I - A) = 0$.

Persamaan $\det(\lambda I - A) = 0$ disebut persamaan karakteristik dari matriks A, dan akar-akar persamaan tersebut, yaitu λ , dinamakan akar-akar karakteristik atau nilai-nilai eigen.

3. Eigenface

Eigenface adalah kumpulan dari eigen vektor yang berasal dari kovarian matrik dengan distribusi acak pada citra wajah dimensi tinggi. Metode ini mentransformasikan citra wajah kedalam sebuah kumpulan karakteristik fitur citra yang dinamakan eigenface. Setiap eigenface menyimpan beberapa bagian dari wajah yang tidak terlihat pada citra yang sesungguhnya. Eigenface didapatkan dengan mengkombinasikan eigen vector dengan citra yang sesungguhnya. 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, $(\Gamma_1, \Gamma_2, \dots, \Gamma_M)$

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

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

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

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

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

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

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

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

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

$$C \times v_i = \lambda_i \times v_i$$

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

$$\mu_l = \sum_{k=1}^M v_{lk} \phi_k$$

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

Tahap 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.

$$\mu_{new} = v \times \Gamma_{new} - \Psi$$

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

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

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

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.

BAB 3

IMPLEMENTASI PROGRAM

1. Eigenvalue dan Eigenvector

Eigenvalue didapatkan dari dekomposisi QR. Misalkan A adalah suatu matriks, dengan melakukan transformasi secara iteratif terhadap matriks A, yaitu dengan melakukan operasi $A_i = Q^T \times A_{i-1} \times Q$, maka pada jumlah iterasi tertentu, matriks A akan berubah menjadi matriks berbentuk segitiga atas. Nilai eigen dari matriks segitiga atas adalah semua elemen diagonal utama pada matriks tersebut. Karena transformasi yang dilakukan adalah similaritas, nilai eigen dari matriks sebelum dan sesudah transformasi bernilai sama. Di samping itu, misalkan $V = Q$ dan kita melakukan iterasi $V_i = V_{i-1} \times Q$, maka saat A berubah menjadi matriks segitiga atas, V merupakan matriks yang vektor-vektor kolomnya merupakan vektor eigen.

2. Eigenface

Pada fungsi ini, digunakan library numpy untuk membantu perhitungan, seperti pengurangan matriks, perkalian matriks, dan mengubah dimensi matriks. Selain itu digunakan juga library cv2 untuk menampilkan gambar eigenface yang diperoleh.

Pada awalnya dilakukan perhitungan kepada list yang berisi matriks yang diperoleh dari dataset. Kemudian matriks-matriks tersebut dimampatkan ukurannya menjadi $N^2 \times 1$. Kemudian dihitung rata-ratanya dengan menjumlahkan semua matriks $N^2 \times 1$ dan dibagi dengan banyak data. Kemudian matriks-matriks $N^2 \times 1$ yang diperoleh di awal dikurangi dengan matriks rata-rata. Kemudian semua matriks digabung menjadi matriks $N^2 \times M$, dengan M merupakan banyak data, yang kolomnya berupa matriks-matriks $N^2 \times 1$ tadi. Agar memudahkan perhitungan, matriks kovarian dihitung secara terbalik. dari matriks kovarian tersebut kemudian dihitung vektor eigennya dengan fungsi eigens yang sudah dibuat sebelumnya. Namun vektor eigen yang didapatkan saat ini bukan eigenface yang diinginkan. Untuk mendapatkan eigenface yang diinginkan, vektor eigen yang telah didapatkan harus di-*dot product* dengan matriks gabungan berukuran $N^2 \times M$ yang telah didapatkan sehingga sekarang, vektor eigen yang didapatkan merupakan eigenface yang diinginkan. Kemudian tiap baris dari eigenface dihitung dengan mengalikan matriks $N^2 \times M$ di awal dengan elemen baris dari vektor eigen yang bersesuaian, sehingga diperoleh eigenface yang berukuran $N^2 \times M$.

3. Tahap pengenalan wajah

Di tahap ini, dilakukan perhitungan terhadap gambar baru. Pertama-tama mengubah gambar menjadi matriks $N \times N$. Kemudian matriks tersebut dimampatkan menjadi matriks $N^2 \times 1$. Matriks tersebut kemudian dikurangi dengan rata-rata dari

matriks yang sudah diperoleh sebelumnya. Kemudian menghitung weight yang didapat dengan mengalikan transpose dari elemen baris yang bersesuaian dengan matriks yang sudah dikurangi sebelumnya. Hasil dari perhitungan tersebut adalah array dengan banyak elemen M.

Selanjutnya adalah menghitung jarak yang dilakukan dengan metode euclidean distance. Weight yang sudah dihitung tadi dikurangi dengan elemen yang bersesuaian. Kemudian, setiap elemen dikuadratkan dan dijumlahkan. Hasil dari penjumlahan tersebut kemudian dilakukan operasi akar kuadrat. Diperoleh array yang berisi jarak-jarak. Kemudian, dilakukan operasi yang digunakan untuk memperoleh nilai minimum dari jarak dan indeks dari matriksnya. Jika nilai tersebut lebih kecil dari ambang batas yang ditentukan, maka gambar yang sesuai dapat ditemukan. Jika lebih besar maka tidak ada wajah yang cocok.

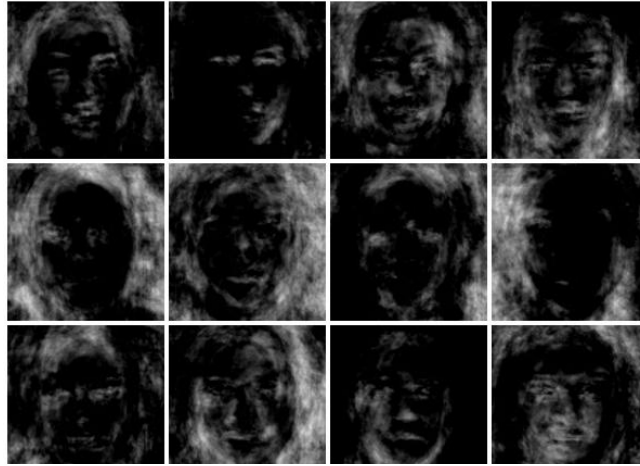
4. GUI

Pada tahap ini, tampilan layout dibuat menggunakan tkinter. Pada tampilan layout akan diminta dua buah input yang pertama berupa folder dataset yang berisi kumpulan gambar yang digunakan sebagai training image, dan yang kedua berupa file gambar yang ingin dikenali dengan format file yang bebas selama merupakan format untuk gambar.

Setelah kita memasukkan input berupa folder dataset dan gambar yang digunakan sebagai training image, Kita bisa memencet tombol process image untuk menjalankan program Face Recognition. Setelah itu program akan mencari file gambar dari dataset yang paling mirip dengan file gambar inputan dan akan menampilkannya di layar.

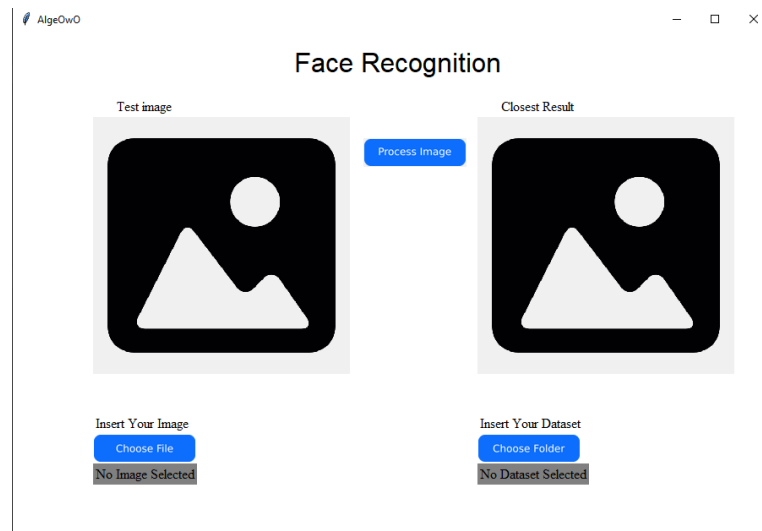
BAB 4 EKSPERIMEN

1. Eigenface



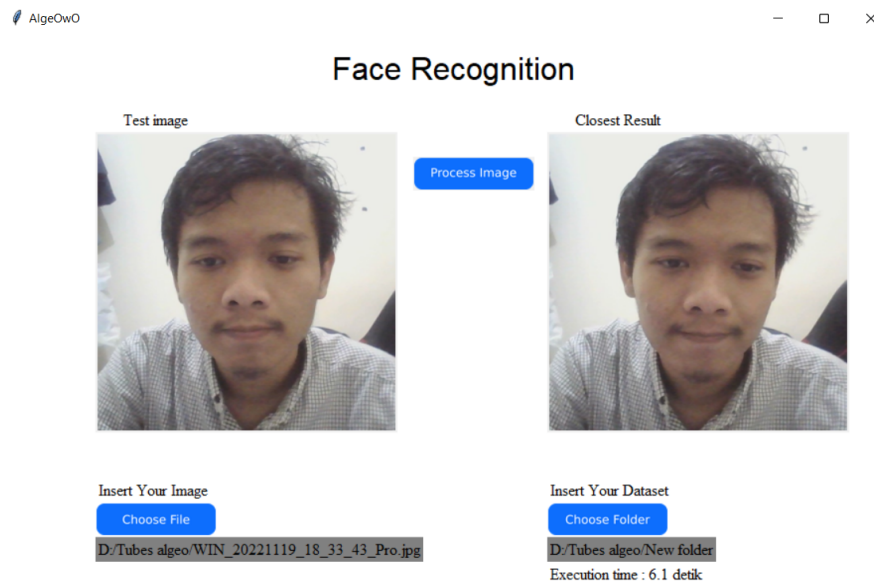
Gambar 4.1 Beberapa eigenfaces yang diperoleh dari perhitungan

2. Tampilan Awal GUI

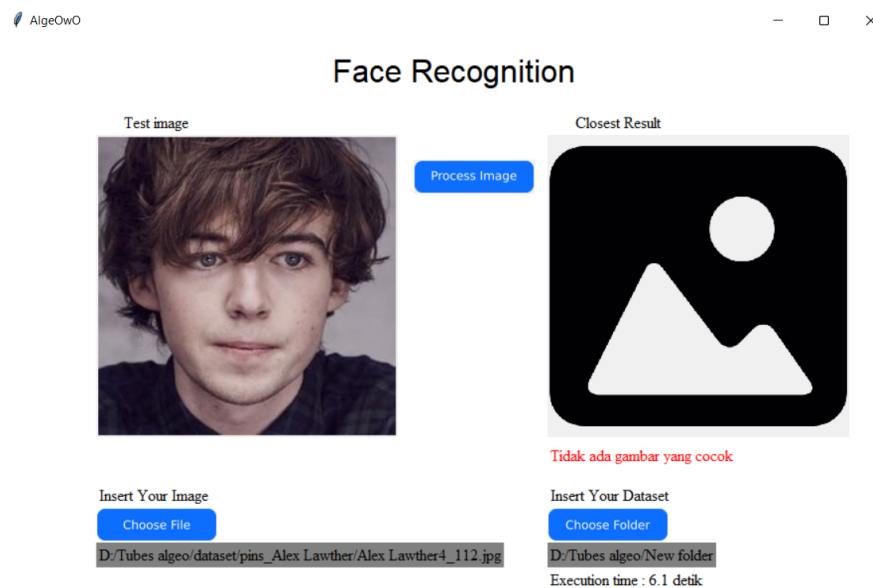


Gambar 4.2.1 Tampilan awal aplikasi

3. Pengenalan Wajah




Gambar 4.3.1 Pengenalan wajah berhasil dilakukan



Gambar 4.3.2 Pengenalan wajah tidak berhasil karena tidak ada wajah yang sesuai


Face Recognition

Test image



Process Image

Closest Result



Tidak ada gambar yang cocok

Insert Your Image

Choose File

D:/Tubes algeo/platelet.jpg

Insert Your Dataset

Choose Folder

D:/Tubes algeo/dataset/pins_barack obama

Execution time : 10.7 detik

Gambar 4.3.3 Pengenalan wajah tidak berhasil karena input gambar bukan merupakan gambar wajah manusia asli

BAB 5

KESIMPULAN, SARAN, REFLEKSI

1. Kesimpulan

Pengenalan wajah dapat dilakukan dengan mengumpulkan dataset foto-foto wajah banyak orang, kemudian matriks dari foto tersebut diolah dengan perhitungan yang menggunakan salah satu materi yang dipelajari di mata kuliah Aljabar Linear dan Geometri, yaitu nilai dan vektor eigen.

2. Saran

Untuk tugas yang akan datang, besar harapan dari anggota kelompok kami akan diadakannya asistensi untuk menanyakan hal-hal yang sekiranya belum dipahami oleh anggota kelompok dibandingkan hanya sekedar *sheet* yang berisi qna.

3. Refleksi

Melalui tugas ini, masing-masing dari kami banyak mengeksplor mengetahui hal-hal baru, seperti nilai dan vektor eigen yang bisa diterapkan untuk pengolahan data wajah, cara membuat GUI, dan lain-lain.

DAFTAR REFERENSI

Watkins, David S. 2002. *Fundamentals of Matrix Computations*. 2nd ed. New York: Wiley Interscience.

<https://saintif.com/perkalian-matriks/>, diakses tanggal 21 November 2022

<https://realpython.com/python-gui-tkinter/>, diakses tanggal 21 November 2022

<https://www.tutorialsteacher.com/python/create-gui-using-tkinter-python>, diakses tanggal 21 November 2022

https://ejournal.undip.ac.id/index.php/berkala_fisika/article/download/4978/4511, diakses tanggal 21 November 2022

<https://onionesquereality.wordpress.com/2009/02/11/face-recognition-using-eigenfaces-and-distance-classifiers-a-tutorial/>, diakses tanggal 21 November 2022

LAMPIRAN

Link repository GitHub:

<https://github.com/kevinjohn01/Algeo02-21042>