

TUGAS BESAR 2 IF2123 ALJABAR LINIER DAN GEOMETRI 2022/2023

Disusun oleh :

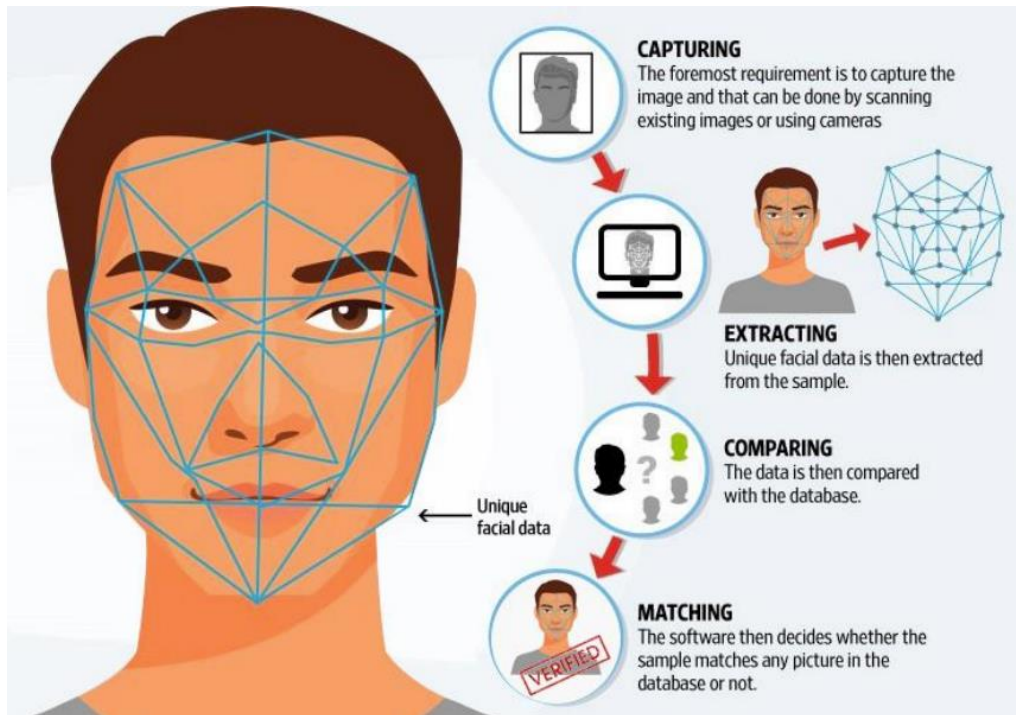
Louis Caesa Kesuma	13521069
Farizki Kurniawan	13521082
Dewana Gustavus Haraka Otang	13521173



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

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 EIGEN FACE

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)

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

$$L = AA^T$$

$$L = \Phi_n \Phi_n^T$$

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:

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

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.

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

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

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

$$\epsilon_k = \Omega - \Omega_k$$

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 II TEORI DASAR

2.1. Perkalian Matrix

Operasi perkalian matrix sesuai namanya adalah operasi yang pada umumnya mengalikan 2 buah matrix. Syarat dari perkalian matrix adalah jumlah kolom matrix pertama harus sama dengan jumlah baris pada matrix kedua. Operasi perkalian matrix terdiri dari mengalikan tiap baris pada matrix pertama dengan tiap kolom pada matrix kedua. Perlu diperhatikan juga bahwa urutan perkalian sangat berpengaruh pada hasil perkalian matrix. Jadi jika A adalah matrix dan B adalah matrix, maka $AB \neq BA$. Operasi perkalian matrix tentunya dapat dipermudah dengan menggunakan library numpy yaitu `matmul`.

2.2. Nilai Eigen

Nilai Eigen adalah nilai yang menyatakan nilai karakteristik dari sebuah matrix yang berukuran $n \times n$. Nilai eigen adalah nilai skalar yang jika dikalikan dengan sebuah vektor, akan bernilai sama dengan sebuah matrix yang dikali dengan vektor yang sama.

2.3. Vektor Eigen

Vektor Eigen adalah sebuah vektor kolom yang apabila dikalikan dengan sebuah Matrix akan menghasilkan vektor lain yang merupakan kelipatan vektor itu sendiri.

2.4. Eigenface

Eigenface adalah sekumpulan vektor eigen yang merepresentasikan ciri citra wajah. Vektor eigennya berasal dari kovarian matrix yang memiliki distribusi probabilitas yang tinggi dan dimensi ruang vektor untuk mengenali sebuah wajah.

BAB III IMPLEMENTASI

Secara garis besar algoritma eigenface yang kami implementasikan pada program ini terdiri dari 2 tahap, yaitu tahap training dan tahap testing.

Pada tahap training terdapat beberapa fungsi yang digunakan:

1. fungsi `vectorsatuan`
Fungsi `vectorsatuan` berfungsi untuk mengubah vector menjadi vektor satuan.
2. fungsi `Euclid_distance`
Fungsi `Euclid_distance` berfungsi untuk mencari Euclidean distance dari 2 buah vektor.
3. fungsi `length`
Fungsi `length` berfungsi untuk mencari length dari sebuah vektor, length tersebut dapat digunakan untuk menghitung proyeksi orthogonal.
4. fungsi `QR`
Fungsi `QR` berfungsi untuk mencari matrix diagonal dari sebuah matrix dan eigenvector dari matrix tersebut.
5. fungsi `getEigenface`
Fungsi `getEigenface` berfungsi untuk mencari eigenface dari masukan eigenvector yang dimasukkan.
6. fungsi `getOmega`
Fungsi `getOmega` berfungsi untuk mencari matrix yang berisikan koefisien dari eigenfaces setiap gambar pada folder.
7. fungsi `training`
Fungsi `training` berfungsi untuk menghasilkan hasil training dari datasetnya, hasil trainingnya berupa K , ψ (rata-rata dari matrix training), Ω (yang berisikan koefisien dari eigenfaces setiap gambar), dan daftar eigenfacenya. Hasil training ini kemudian digunakan pada tahap testing.

Kemudian pada tahap testing hanya terdapat satu fungsi yang digunakan:




- A. fungsi `indeks_gambar_terdekat`

Fungsi `indeks_gambar_terdekat` berfungsi untuk mencari index gambar pada folder yang dimana gambar tersebut jika Omeganya diselisihkan dengan Omega gambar testing akan bernilai terkecil dari seluruh gambar di folder.




BAB IV EKSPERIMEN

4.1. Variasi terhadap ukuran gambar (folder berisi 300 gambar)

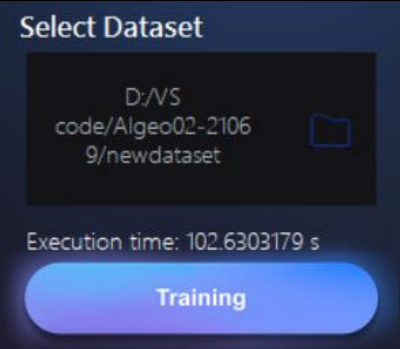


a. 128 x 128

Execution time		
Hasil analisa		

b. 256 x 256

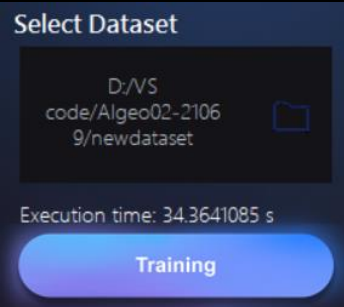


Execution time		
Hasil analisa		

c. 512 x 512



Execution time		
Hasil analisa		

4.2. Variasi terhadap banyak dataset (sizenya 256 x 256)



a. 300 gambar

Execution time		
Hasil analisa		

b. 200 gambar


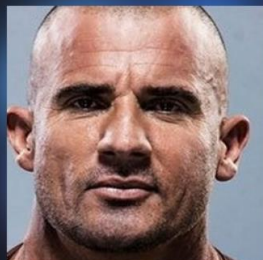
Execution time	<div> <div>Select Dataset</div> <div> D:\VS code\Algeo02-2106 9/newdataset200 </div> <div>Execution time: 20.674642000000006 s</div> <div>Training</div> </div>	
Hasil analisa	<div>Test image</div> 	<div>Closest result</div> 

c. 100 gambar

Execution time	<div> <div>Select Dataset</div> <div> D:\VS code\Algeo02-2106 9/newdataset100 </div> <div>Execution time: 8.2530231000000007 s</div> <div>Training</div> </div>	
Hasil analisa	<div>Test image</div> 	<div>Closest result</div> 



4.3. Hasil pemakaian recognition dengan kamera (size 256x256, 300 gambar)

a. Dewana


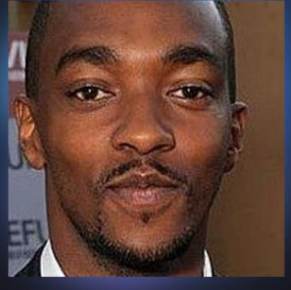
Execution time	<div> <div>Select Dataset</div> <div> <div>and</div> <div>Kerja/tubes/algeo/A</div> <div>Igeo02-21069/datas</div> </div> <div>Time elapsed: 28.548807899999996 s</div> <div>Training</div> </div>	
Hasil analisa	<div>Test image</div> 	<div>Closest result</div> 

4.4. Variasi terhadap besar toleransi (size 256x256, 300 gambar)



a. Tolerance 10% dari rata-rata

Execution time	<div> <div>Select Dataset</div> <div> <div>and</div> <div>Kerja/tubes/algeo/A</div> <div>Igeo02-21069/datas</div> </div> <div>Time elapsed: 28.548807899999996 s</div> <div>Training</div> </div>	
Hasil analisa	<div>Test image</div> 	<div>Closest result</div> 

b. Tolerance 8% dari rata-rata

Execution time	<div> <div>Select Dataset</div> <div> <div>and</div> <div>Kerja/tubes/algeo/A</div> <div>lgeo02-21069/datas</div> </div> <div>Time elapsed: 28.548807899999996 s</div> <div>Training</div> </div>	
Hasil analisa	<div>Test image</div> 	<div>Closest result</div> 

c. Tolerance 5% dari rata-rata

Execution time	<div> <div>Select Dataset</div> <div> <div>and</div> <div>Kerja/tubes/algeo/A</div> <div>lgeo02-21069/datas</div> </div> <div>Time elapsed: 28.548807899999996 s</div> <div>Training</div> </div>	
Hasil analisa	<div>Test image</div> 	<div>Closest result</div> 

BAB V KESIMPULAN, SARAN, DAN REFLEKSI

5.1. Kesimpulan

Face recognition sangatlah berguna bagi teknologi zaman ini, banyak sekali teknologi yang memiliki fitur tersebut. Dari hasil yang kami capai, face recognition dengan menggunakan metode eigenface lumayan mudah untuk diimplementasikan namun terkadang terdapat ketidaksesuaian dalam perhitungan yang menyebabkan gambar tersebut terkesan sama pada program namun terlihat tidak sama jika dilihat dengan mata kita sendiri.

5.2. Saran

Pilihlah gambar yang terlihat jelas dimata, karena kemungkinan besar ketidaksesuaian perhitungannya dapat dikurangi. Coba lakukan percobaan dimana segala fungsi yang kemungkinan memperlambat program dihilangkan, kemungkinan besar program dapat dijalankan dengan baik dan waktu yang dipakai lebih sedikit.

5.3. Refleksi

Pilihlah metode yang efisien untuk program yang akan dibuat, jika tidak efisien maka setidaknya pikirkanlah metode yang mudah untuk diimplementasikan dan sesuai dengan programnya.

REFERENSI

edunex.itb.ac.id (Diakses pada tanggal 19 November 2022)

www.geeksforgeeks.com. (2021, 24 September). ML| Face Recognition Using Eigenface (PCA Algorithm). Diakses pada tanggal 1 November 2022.

<https://stackoverflow.com/questions/67721086/how-to-resize-live-video-in-python-cv2-using-set>. (2021, 20 Juni). How to resize live video in python cv2 using .set()?. Diakses pada tanggal 20 November 2022.

<https://solarianprogrammer.com/2018/04/21/python-opencv-show-video-tkinter-window/>. (2018, 21 April). Python OpenCV – show a video in a Tkinter window. Diakses pada tanggal 20 November 2022.

[Python GUI Programming With Tkinter – Real Python](#). (2022, 30 Maret). Python GUI Programming With Tkinter. Diakses pada tanggal 17 November 2022.

LAMPIRAN

Github : <https://github.com/DewanaGustavus/Algeo02-21069.git>

Youtube : bit.ly/VideoDemoDewaLoKi