LAPORAN TUGAS BESAR 2 IF2123 ALJABAR LINIER DAN GEOMETRI:
APLIKASI NILAI EIGEN DAN EIGENFACE PADA PENGENALAN WAJAH (FACE RECOGNITION)

Oleh:

Muhammad Raihan Iqbal 13518134

Muhammad Dhiwaul Akbar 13521158

Asdfasf asdfasf

PROGRAM STUDI TEKNIK INFORMATIKA

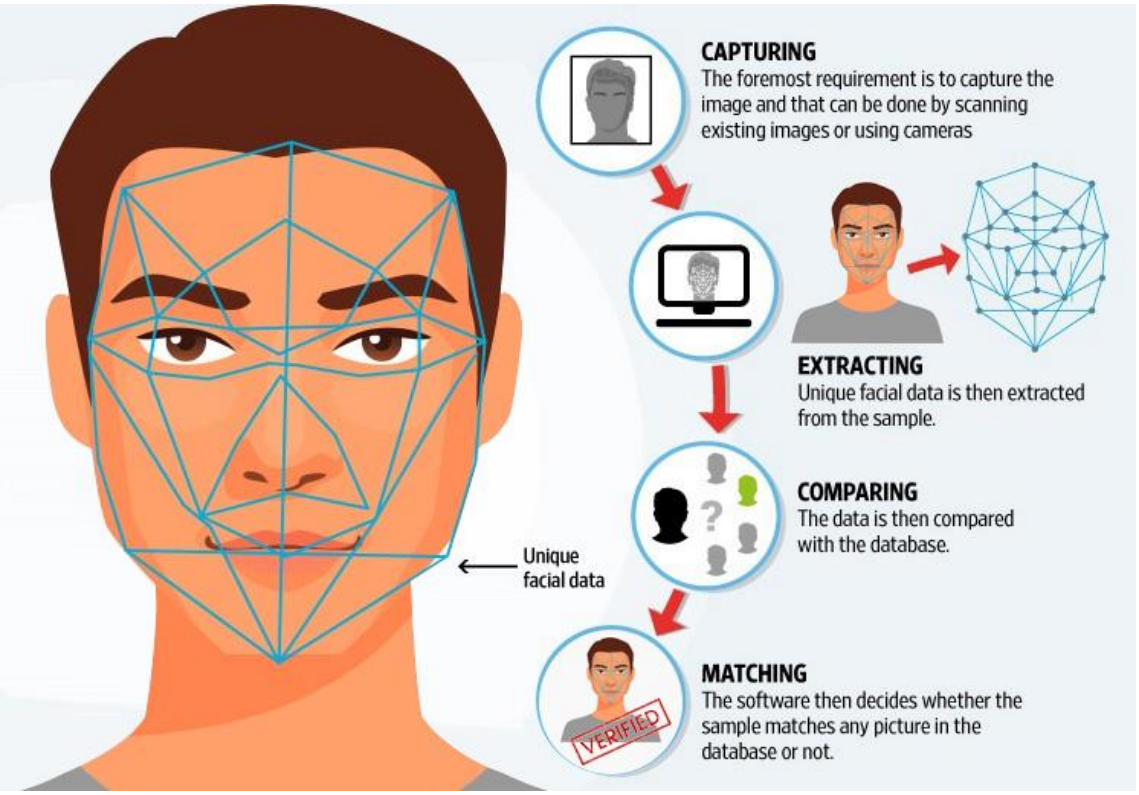
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022

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 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 Eigenface

1. Langkah pertama adalah menyiapkan data dengan membuat suatu himpunan S yang terdiri dari seluruh training image, (Γ1, Γ2, …, ΓM)

   S = (Γ1, Γ2, …, ΓM)

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

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

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

   $$\phi i = \Gamma i - \Psi$$

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

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

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

   $$C \text{ x } vi = \lambda i \text{ x } vi$$

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

$$\mu_i = \sum_{k=1}^{M} v_{ik} \phi_v \quad (6)$$

$$l = 1, \ldots, M$$

Tahapan Pengenalan wajah :

1. Sebuah image wajah baru atau test face (Γnew) akan dicoba untuk dikenali, pertama terapkan cara pada tahapan pertama perhitungan eigenface untuk mendapatkan nilai eigen dari image tersebut.

$$\mu new = v \text{ x } \Gamma new - \Psi \quad (7)$$

$$\Omega = \mu 1 \, , \mu 2 \, , \ldots , \mu M$$

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.

Kode program untuk ekstraksi fitur dapat dibaca pada artikel ini: Feature extraction and
similar image search with OpenCV for newbies, pada laman:
https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774

Nilai batas kemiripan citra test face dapat ditentukan oleh pembuat program melalui percobaan.

Berikut merupakan referensi pengenalan wajah dengan metode eigenface:
https://jurnal.untan.ac.id/index.php/jcskommipa/article/download/9727/9500
https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/

Bab 2
Teori Singkat

1. Perkalian Matriks

Pada tugas besar kali ini , perkalian vektor yang kami lakukan adalah untuk mencari nilai kovarian matrix serta nilai eigen dan vektor eigen dengan cara menggunakan library numpy. Secara teori , perkalian yang dimaksud adalah perkalian dot pada matrix yang menghasilkan matrix kembali . dengan metode seperti yang dibawah ini, Jika $\mathbf{A}$ adalah matriks berukuran *m* × *n* dan $\mathbf{B}$ adalah matriks berukuran $n \times p$, dengan elemen-elemen sebagai berikut.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

Hasil perkalian kedua matriks tersebut, $\mathbf{C} = \mathbf{AB}$ (dinyatakan tanpa menggunakan tanda kali atau titik), adalah sebuah matriks berukuran $m \times p$.

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{pmatrix}$$

dengan setiap entri pada matriks $\mathbf{C}$ didefinisikan sebagai

$$c_{ij} = a_{i1}b_{1j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj},$$

untuk nilai *i* = 1, … , *m* dan nilai $i = 1, \ldots, p$. Dengan kata lain, entri c_{ij} adalah hasil yang didapatkan dengan mengalikan secara berpasang-pasangan entri di baris ke-i matriks $\mathbf{A}$ dengan entri di kolom ke-j matriks $\mathbf{B}$, lalu menjumlahkan semua hasil perkalian ini. Intepretasi lain dari proses ini, entri c_{ij} adalah hasil perkalian

titik baris ke-i matriks $\mathbf{A}$ dengan kolom ke-j matriks $\mathbf{B}$. Dengan demikian, $\mathbf{AB}$ juga dapat ditulis sebagai berikut.

$$\mathbf{C} = \begin{pmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & a_{11}b_{12} + \cdots + a_{1n}b_{n2} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ a_{21}b_{11} + \cdots + a_{2n}b_{n1} & a_{21}b_{12} + \cdots + a_{2n}b_{n2} & \cdots & a_{21}b_{1p} + \cdots + a_{2n}b_{np} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & a_{m1}b_{12} + \cdots + a_{mn}b_{n2} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{pmatrix}$$

2. Nilai Eigen

Secara matematis, misalnya kita punya matriks A, dengan vektor eigen dari matriks tersebut adalah x, dan nilai eigennya λ. Maka hubungan antara vektor dan nilai tersebut yaitu:

$Ax = \lambda x$

Nilai eigen dari vektor A dapat dicari dengan menyelesaikan persamaan berikut:

$\text{Det } |\lambda I - A| = 0$

3. Vektor Eigen

Persamaan Ax = λx , kedua ruas dikalikan dengan matriks identitas menjadi persamaan dibawah

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

Atau dalam bentuk persamaan linier dapat dituliskan :

$$(a_{11} - \lambda)x_1 + a_{12}x_2 = 0$$
$$a_{21}x_1 + (a_{22} - \lambda)x_2 = 0$$

Persamaan diatas adalah sistem persamaan linier homogen, vektor dalam ruang Rn yang tidak nol didapatkan jika dan hanya jika persamaan tersebut mempunyai solusi non trivial untuk nilai eigen yang sesuai

4. Eigenface

Menggunakan persamaan berikut

$$F = F_m + \sum_{i=1}^{n} \alpha_i F_i$$

- F = wajah baru,
- Fm = wajah rata-rata,
- Fi = sebuah eigenface ,
- Alpha_i = skalar yang bisa kita pilih untuk mebuat wajah baru

Bab 3
Implementasi Pustaka dan Program

Pada aplikasi ini, terdapat beberapa pustaka dan program yang kami gunakan, diantaranya yaitu:

1. PyQt6

   Pustaka ini digunakan untuk membentuk GUI dari aplikasi yang diimplementasikan.
2. Qt Designer

   Program ini digunakan untuk melakukan desain awal terhadap aplikasi GUI yang akan digunakan dan dimodifikasi lebih lanjut.
3. pyuic6

   Pustaka ini digunakan melakukan konversi dari file .ui yang dihasilkan oleh Qt Designer ke dalam bentuk .py agar dapat diolah oleh program utama yang dibuat.

Bab 4
Eksperimen

Kami belum melakukan eksperimen.

Bab 5
Kesimpulan, Saran, dan Refleksi

1. Kesimpulan

Program yang kami buat dapat dijalankan, dan dapat memenuhi spek-spek berikut:

- Dalam melakukan pembacaan terhadap input-input yang telah diberikan.
- Terdapat fitur untuk melakukan kalkulasi nilai eigen untuk digunakan pada perhitungan eigenface.

2. Saran

Melakukan optimalisasi dari kode yang telah dikembangkan sebelumnya.

3. Refleksi

- Perlu komunikasi lebih dengan anggota kelompok
- Terdapat salah satu anggota yang tidak muncul dalam kerja kelompok, sehingga kerja tugas besar 2 ini hanya dikerjakan oleh 2 orang.

Referensi

https://stackoverflow.com/questions/41587541/pyqt-assign-to-a-button-the-ability-to-choose-a-directory

https://doc.qt.io/qtforpython-5/PySide2/QtGui/QPixmap.html#more

Eigenfaces for Face recognition (opengenus.org)

http://kseminar.staff.ipb.ac.id/files/2013/02/EIGEN-VALUE-VEKTOR.pdf

https://dosen.itats.ac.id/anitateku/wp-content/uploads/sites/17/2015/11/4.-NILAI-EIGEN-dan-VEKTOR-EIGEN.pdf

Lampiran

Link ke laman dokumen:
https://docs.google.com/document/d/11kV0STaxiamU5L_AhCu6okfO2MMQebtg4SJ-BJmvKrE/edit?usp=sharing

Link Repository:
https://github.com/raihaniqbal24/Algeo02-18134

Link video youtube: