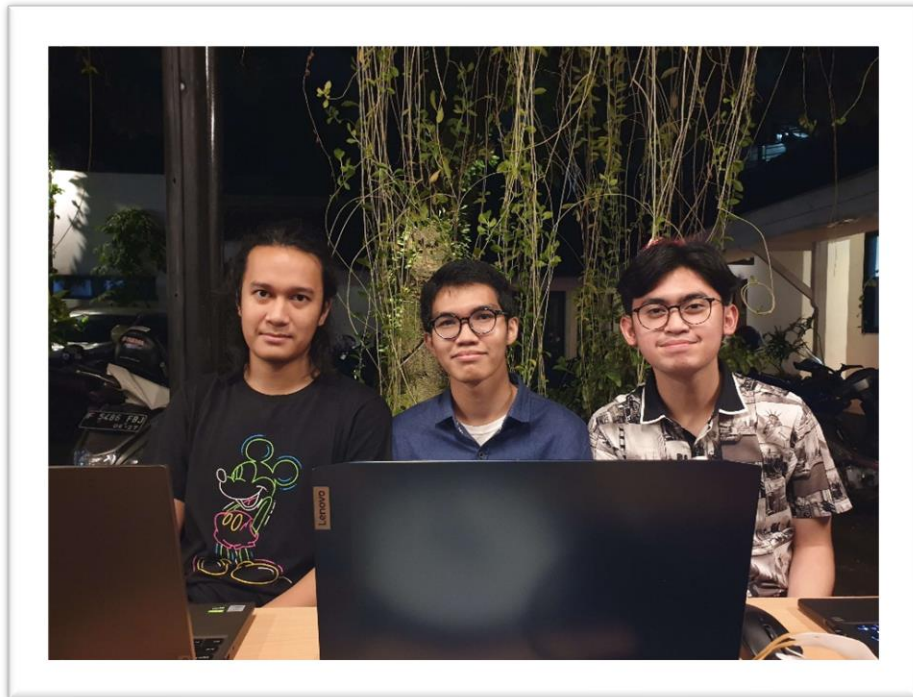


LAPORAN TUGAS BESAR

IF2123 ALJABAR LINEAR DAN GEOMETRI

Aplikasi Nilai Eigen dan EigenFace pada Pengenalan Wajah (*Face Recognition*)



Disusun oleh

Christian Albert Hasiholan 13521078

Noel Christoffel Simbolon 13521096

Ammar Rasyad Chaeroel 13521136

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

2022

DAFTAR ISI

DAFTAR ISI.....	i
BAB I DESKRIPSI MASALAH.....	1
1.1 Tujuan.....	1
1.2 Spesifikasi Tugas.....	1
BAB II TEORI SINGKAT.....	2
2.1 Perkalian Matriks	2
2.2 Nilai dan Vektor Eigen.....	2
2.3 <i>Eigenface</i>	2
BAB III IMPLEMENTASI.....	4
3.1 <i>Tech Stack</i>	4
3.2 Garis Besar Algoritma.....	4
BAB IV EKSPERIMEN	5
4.1 [Judul Subbab].....	5
4.2 [Judul Subbab].....	5
BAB V PENUTUP.....	6
5.1 Simpulan.....	6
5.2 Saran.....	6
5.3 Refleksi.....	6
DAFTAR REFERENSI.....	7
LAMPIRAN.....	8
Link Repository	8

BAB I

DESKRIPSI MASALAH

1.1 Tujuan

Tujuan dari tugas besar ini adalah sebagai berikut.

1. Membuat program *face recognition* dalam bahasa Python dengan memanfaatkan sejumlah *library* di OpenCV atau *library* pemrosesan gambar lainnya.
2. Menggunakan program tersebut untuk melakukan pengenalan wajah melalui tahap *training* dan tahap pencocokan.

1.2 Spesifikasi Tugas

Spesifikasi dari program yang dibuat adalah 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 satu 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 256×256 .
8. Hasil *training* dapat disimpan, tetapi tetap wajib dapat dilakukan *training* kembali jika diperlukan *user*.

BAB II

TEORI SINGKAT

2.1 Perkalian Matriks

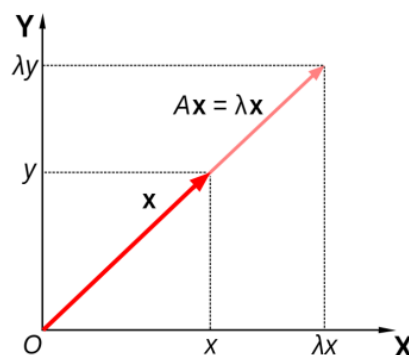
Perkalian matriks merupakan perkalian elemen-elemen pada matriks berdasarkan baris dan kolomnya. Misal dilakukan perkalian matriks A dengan matriks B, maka syarat berlakunya perkalian matriks hanya ketika kolom matriks A sama besarnya dengan baris matriks B. Kemudian dilakukan perkalian elemen dari baris matriks A dengan elemen dari kolom matriks B, dan dijumlahkan sehingga mendapat elemen untuk baris dan kolom tersebut pada matriks hasilnya.

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

2.2 Nilai dan Vektor Eigen

Vektor eigen adalah vektor karakteristik dari suatu matriks A berukuran $n \times n$. Ketika vektor eigen dikalikan dengan matriks tersebut, maka akan menghasilkan vektor lain yang merupakan kelipatan dari vektor itu sendiri. Kelipatan dari vektor itu merupakan hasil perkalian sebuah skalar dengan vektor eigen, nilai skalar itu disebut juga dengan nilai eigen.

$$A\mathbf{x} = \lambda\mathbf{x}$$



2.3 Eigenface

Eigenface merupakan salah satu metode pengenalan wajah (*face recognition*). Metode *eigenface* didasari dengan vektor *eigen* dan nilai *eigen* yang digunakan untuk persoalan

computer vision. Vektor *eigen* diperoleh dari matriks kovarian yang berasal dari sekumpulan *training image*. Kumpulan *training image* yang diterima program kemudian dibentuk menjadi himpunan basis yang berbentuk matriks. Dari matriks tersebut kemudian dapat diperoleh matriks kovarian. Setelah matriks kovarian didapat maka vektor *eigen* dapat ditentukan. Maka tiap *training image* dapat ditentukan kombinasi linearnya berdasarkan *eigenface* yang diperoleh.



BAB III IMPLEMENTASI

3.1 *Tech Stack*

1. OpenCV

OpenCV digunakan untuk mengekstrak *training* dan *testing image* yang akan digunakan program. Karena *face recognition* tidak memerlukan image berwarna, maka semua image diubah warnanya menjadi *grayscale* menggunakan fungsi pada OpenCV juga. Pada spesifikasi hanya terdapat ukuran minimal *image* yaitu 256×256 , maka untuk membuat ukuran semua *image* sama digunakan fungsi *resize* sehingga ukuran tiap *image* sama.

2. NumPy/CuPy

NumPy dan CuPy memiliki fungsi yang sama, yaitu untuk mempermudah perhitungan pada program. Dalam program ini terdapat banyak operasi matriks, karena matriks memiliki ukuran besar maka library tersebut digunakan sehingga program tidak memakan waktu dan memori yang besar.

3. CustomTkinter

CustomTkinter digunakan untuk membuat GUI program ini. CustomTkinter dipakai sehingga diperoleh GUI yang lebih modern daripada Tkinter.

3.2 *Garis Besar Algoritma*

1. Program menerima *dataset* untuk proses *training*
2. Dilakukan *resizing* pada tiap *training image* sehingga berukuran 256×256
3. Ubah *training image* menjadi vektor kemudian disatukan menjadi matriks
4. Cari nilai rata-rata atau *mean* dari matriks tadi.
5. *Normalize* matriks dengan mengurangi tiap kolom dengan *mean*.
6. Hitung matriks kovarian dengan mengalikan *transpose normalize* matriks dengan *normalize* matriks, sehingga diperoleh matriks kovarian tereduksi.
7. Tentukan *eigenvalue* dan *eigenvector* dari matriks kovarian tereduksi.
8. Kalikan *normalize* matriks dengan *eigenvector* sehingga diperoleh *eigenfaces*.
9. Tentukan kombinasi linear untuk tiap *normalize training image* berdasarkan *eigenfaces*.
10. Koefisien dari kombinasi linear sebelumnya dibentuk menjadi vektor.
11. Program menerima *testing image*.
12. Ubah ukurannya menjadi sama seperti *training image*, lalu jadikan vektor.
13. *Normalize* vektor *testing image*.
14. Proyeksi vektor *testing image* yang telah di-*normalize* ke *eigenspace* agar diperoleh kombinasi linear dari *eigenfaces*.
15. Dari proyeksi tersebut diperoleh vektor koefisien.
16. Cari *euclidean distance* terkecil antara vektor koefisien *testing image* dengan kumpulan vektor koefisien *training image*.
17. Jika *euclidean distance* kurang dari nilai batas maka diperoleh *training image* yang paling mirip dengan *testing image*.

BAB IV

EKSPERIMEN

4.1 [Judul Subbab]

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

4.2 [Judul Subbab]

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Sed blandit libero volutpat sed cras ornare arcu dui vivamus. Vulputate odio ut enim blandit volutpat. Nunc mattis enim ut tellus. Porta nibh venenatis cras sed. Eu feugiat pretium nibh ipsum consequat nisl vel pretium. A lacus vestibulum sed arcu non odio. Ut eu sem integer vitae. Tempus quam pellentesque nec nam. Morbi leo urna molestie at elementum. Ullamcorper velit sed ullamcorper morbi tincidunt ornare massa. Fermentum posuere urna nec tincidunt. Lacus vel facilisis volutpat est velit egestas. Amet mauris commodo quis imperdiet massa tincidunt nunc pulvinar. Morbi enim nunc faucibus a pellentesque sit amet porttitor. Turpis tincidunt id aliquet risus feugiat in ante metus (Akrou, Feriani, & Akrou, 2019).

BAB V

PENUTUP

5.1 Simpulan

1. Program pengenalan wajah telah diimplementasikan dan dibuat aplikasinya, seperti yang telah dipaparkan dengan lanjut detailnya pada bab III.
2. Program telah digunakan untuk melakukan pengenalan wajah, seperti yang telah dieksperimenkan pada bab IV.

5.2 Saran

1. Diperbolehkan menggunakan berbagai fungsi pada library untuk memperlancar program.
2. Deadline tidak bertumpuk dengan tugas besar lain.

5.3 Refleksi

Kami, sebagai satu kelompok, merasa puas selama proses pengerjaan tugas besar ini karena kami belajar banyak hal mengenai aplikasi nilai *eigen* dan *eigenface* untuk pengenalan wajah. Namun, karena ditambah dengan beban dari tugas-tugas besar mata kuliah lain, kami merasa sedikit tertekan. Dari segi kekurangan, kerja sama kami dapat di-*improve* sebagai suatu tim. Selain itu, komunikasi antaranggota juga masih kurang optimal sehingga menyebabkan ketidakmaksimalan pada penyelesaian dan pengembangan program.

DAFTAR REFERENSI

- Ahn, L. v., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008, September 12). reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895), 1465-1468. doi:10.1126/science.1160379
- Akrout, I., Feriani, A., & Akrou, M. (2019, April 18). Hacking Google reCAPTCHA v3 using Reinforcement Learning. *arXiv preprint*. Retrieved from <https://arxiv.org/abs/1903.01003>
- Dzieza, J. (2019, February 1). *Why CAPTCHAs have gotten so difficult*. Retrieved from The Verge: <https://www.theverge.com/2019/2/1/18205610/google-captcha-ai-robot-human-difficult-artificial-intelligence#AI%20in%20Pop%20Culture>
- Wirth, N. (2004). *Algorithms and Data Structures* (Oberon ed.). Moscow.

LAMPIRAN

Link Repository

<https://github.com/ChrisAlberth/Algeo02-21078.git>