

LAPORAN TUGAS BESAR

Aplikasi Nilai Eigen dan Vektor Eigen dalam Kompresi Gambar

Ditujukan untuk memenuhi salah satu tugas besar mata kuliah IF2123 Aljabar Linear dan Geometri pada Semester I Tahun Akademik 2021/2022



Disusun oleh:
Kelompok 22 LUCKY SWANDI

Adiyansa Prasetya Wicaksana	13520044
Samuel Christopher Swandi	13520075
Monica Adelia	13520096

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

2021

Daftar Isi

Daftar Isi	1
BAB 1 Deskripsi Masalah	1
BAB 2 Teori Singkat	5
BAB 3 Implementasi	7
BAB 4 Eksperimen	10
BAB 5	23
Kesimpulan	23
Saran	23
Refleksi	23
Referensi	25

BAB 1

Deskripsi Masalah

Gambar adalah suatu hal yang sangat dibutuhkan pada dunia modern ini. Kita seringkali berinteraksi dengan gambar baik untuk mendapatkan informasi maupun sebagai hiburan. Gambar digital banyak sekali dipertukarkan di dunia digital melalui file-file yang mengandung gambar tersebut. Seringkali dalam transmisi dan penyimpanan gambar ditemukan masalah karena ukuran file gambar digital yang cenderung besar.

Kompresi gambar merupakan suatu tipe kompresi data yang dilakukan pada gambar digital. Dengan kompresi gambar, suatu file gambar digital dapat dikurangi ukuran filenya dengan baik tanpa mempengaruhi kualitas gambar secara signifikan. Terdapat berbagai metode dan algoritma yang digunakan untuk kompresi gambar pada zaman modern ini.



Three levels of JPEG compression. The left-most image is the original. The middle image offers a medium compression, which may not be immediately obvious to the naked eye without closer inspection. The right-most image is maximally compressed.

Gambar 1. Contoh kompresi gambar dengan berbagai tingkatan

Sumber : [Understanding Compression in Digital Photography \(lifewire.com\)](http://lifecycle.com/Understanding%20Compression%20in%20Digital%20Photography)

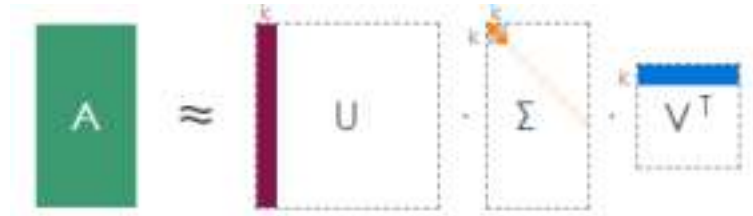
Salah satu algoritma yang dapat digunakan untuk kompresi gambar adalah algoritma SVD (Singular Value Decomposition). Algoritma SVD didasarkan pada teorema dalam aljabar linier yang menyatakan bahwa sebuah matriks dua dimensi dapat dipecah menjadi hasil perkalian dari 3 sub-matriks yaitu matriks ortogonal U, matriks diagonal S, dan transpose dari matriks ortogonal V. Dekomposisi matriks ini dapat dinyatakan sesuai persamaan berikut.

$$A_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T$$

Gambar 1. Algoritma SVD

Matriks U adalah matriks yang kolomnya terdiri dari vektor eigen ortonormal dari matriks AA^T . Matriks ini menyimpan informasi yang penting terkait baris-baris matriks awal, dengan informasi terpenting disimpan di dalam kolom pertama. Matriks S adalah matriks diagonal yang berisi akar dari nilai eigen matriks U atau V yang terurut menurun.

Matriks V adalah matriks yang kolomnya terdiri dari vektor eigen ortonormal dari matriks $A^T A$. Matriks ini menyimpan informasi yang penting terkait kolom-kolom matriks awal, dengan informasi terpenting disimpan dalam baris pertama.



Gambar 2. Ilustrasi Algoritma SVD dengan rank k

Dapat dilihat di gambar di atas bahwa dapat direkonstruksi gambar dengan banyak *singular values* k dengan mengambil kolom dan baris sebanyak k dari U dan V serta *singular value* sebanyak k dari S atau Σ terurut dari yang terbesar. Kita dapat mengaproksimasi suatu gambar yang mirip dengan gambar aslinya dengan mengambil k yang jauh lebih kecil dari jumlah total *singular value* karena kebanyakan informasi disimpan di *singular values* awal karena *singular values* terurut mengecil. Nilai k juga berkaitan dengan rank matriks karena banyaknya *singular value* yang diambil dalam matriks S adalah rank dari matriks hasil, jadi dalam kata lain k juga merupakan rank dari matriks hasil. Maka itu matriks hasil rekonstruksi dari SVD akan berupa informasi dari gambar yang terkompresi dengan ukuran yang lebih kecil dibanding gambar awal.

Pada kesempatan kali ini, diminta untuk membuat website kompresi gambar sederhana dengan menggunakan algoritma SVD.

SPESIFIKASI TUGAS

Buatlah program kompresi gambar dengan memanfaatkan algoritma SVD dalam bentuk website lokal sederhana. Spesifikasi website adalah sebagai berikut:

1. Website mampu menerima *file* gambar beserta *input* tingkat kompresi gambar (dibebaskan formatnya).
2. Website mampu menampilkan gambar *input*, *output*, *runtime* algoritma, dan persentase hasil kompresi gambar (perubahan jumlah pixel gambar).
3. File *output* hasil kompresi dapat diunduh melalui website.
4. Kompresi gambar tetap mempertahankan warna dari gambar asli.
5. **(Bonus)** Kompresi gambar tetap mempertahankan transparansi dari gambar asli, misal untuk gambar png dengan *background* transparan.
6. Bahasa pemrograman yang boleh digunakan adalah Python, Javascript, dan Go.
7. Penggunaan *framework* untuk *back end* dan *front end* website dibebaskan. Contoh *framework* website yang bisa dipakai adalah Flask, Django, React, Vue, dan Svelte.
8. Kalian dapat menambahkan fitur fungsional lain yang menunjang program yang anda buat (unsur kreativitas diperbolehkan/dianjurkan).

9. Program harus modular dan mengandung komentar yang jelas.
10. Diperbolehkan menggunakan *library* pengolahan citra seperti OpenCV2, PIL, atau image dari Go. **Dilarang** menggunakan *library* perhitungan SVD dan *library* pengolahan eigen yang sudah jadi.

BAB 2

Teori Singkat

A. Perkalian Matriks

Perkalian matriks adalah suatu operasi biner dari dua matriks yang menghasilkan sebuah matriks. Agar dua buah matriks dapat dikalikan, jumlah kolom matriks pertama harus sama dengan jumlah baris B. Perkalian dua matriks memiliki rincian seperti berikut:

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

Misalnya, $A = [a_{ij}]$ dan $B = [b_{ij}]$, maka,

$$C = A \times B = [C_{ij}]$$

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$$

Terdapat beberapa sifat pada perkalian matriks, antara lain:

- a) $A \times 0 = 0 \times A = 0$
- b) $A \times I = I \times A = A$
- c) $(A \times B) \times C = A \times (B \times C)$
- d) $A \times (B + C) = (A \times B) + (A \times C)$
- e) $c(A \times B) = (c \times A) \times B$

B. Nilai eigen

Nilai eigen atau *eigenvalues* menyatakan nilai karakteristik dari sebuah matriks yang berukuran $n \times n$. Sebuah matriks A berukuran $n \times n$, dihitung sebagai berikut:

$$Ax = \lambda x$$

$$IAx = \lambda Ix$$

$$Ax = \lambda Ix$$

$$(\lambda I - A)x = 0$$

Supaya λ menjadi nilai eigen dari A , maka x harus merupakan vektor tak nol. Sistem persamaan diatas akan mempunyai penyelesaian jika dan hanya jika koefisien matriks $(\lambda I - A)$ mempunyai determinan nol. Oleh karena itu, kita bisa merumuskannya menjadi sebuah teorema sebagai berikut:

$$\det(\lambda I - A) = 0$$

Persamaan ini disebut persamaan karakteristik dari matriks A dan akar-akar persamaan tersebut, yaitu λ , dinamakan akar-akar karakteristik atau nilai-nilai eigen.

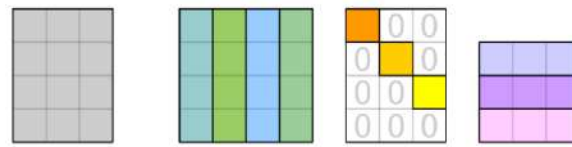
C. Vektor eigen

Vektor eigen adalah vektor kolom bukan nol yang bila dikalikan dengan suatu matriks berukuran $n \times n$ akan menghasilkan vektor lain yang memiliki nilai kelipatan

dari vektor Eigen itu sendiri. Vektor eigen dapat diperoleh dengan mencari nilai eigen matriksnya terlebih dahulu. Lalu nilai eigen (λ) tersebut disubstitusi ke dalam rumus $(\lambda I - A)x = 0$. Vektor yang didapat sebagai penyelesaian persamaan tersebutlah yang disebut vektor eigen.

D. Matriks SVD

Singular value decomposition (SVD) adalah salah satu metode pemfaktoran untuk matriks non-bujursangkar. SVD adalah suatu teknik untuk mendekomposisi matriks berukuran apa saja untuk mempermudah pengolahan data. SVD memfaktorkan matriks A berukuran $m \times n$ menjadi matriks U , Σ , dan V sedemikian sehingga $A = U \Sigma V^T$.



$$\begin{matrix} \mathbf{M} & = & \mathbf{U} & \mathbf{\Sigma} & \mathbf{V}^* \\ m \times n & & m \times m & m \times n & n \times n \end{matrix}$$

Langkah-langkah SVD mendekomposisi $A_{m \times n}$ menjadi U , Σ , dan V adalah sebagai berikut:

- Mencari matriks U dengan menghitung nilai-nilai eigen dari AA^T . Lalu, menentukan vektor-vektor eigen u_1, u_2, \dots, u_m yang berkoresponden dengan nilai-nilai eigen dari AA^T . Normalisasi u_1, u_2, \dots, u_m dengan cara setiap komponen vektornya dibagi dengan panjang vektor.
- Mencari matriks V dengan menghitung nilai-nilai eigen dari $A^T A$ lalu menentukan nilai-nilai singularnya. Kemudian, menentukan vektor-vektor eigen v_1, v_2, \dots, v_n yang berkoresponden dengan nilai-nilai eigen dari $A^T A$. Normalisasi v_1, v_2, \dots, v_n dengan cara setiap komponen vektornya dibagi dengan panjang vektor. Lalu transposekan matriks V sehingga didapat V^T .
- Bentuk matriks Σ berukuran $m \times n$ dengan elemen-elemen diagonalnya adalah nilai-nilai singular dari matriks A dengan susunan dari besar ke kecil. Nilai singular di dalam Σ adalah akar pangkat dua dari nilai-nilai eigen yang tidak nol dari $A^T A$.
- Maka, $A = U \Sigma V^T$

BAB 3

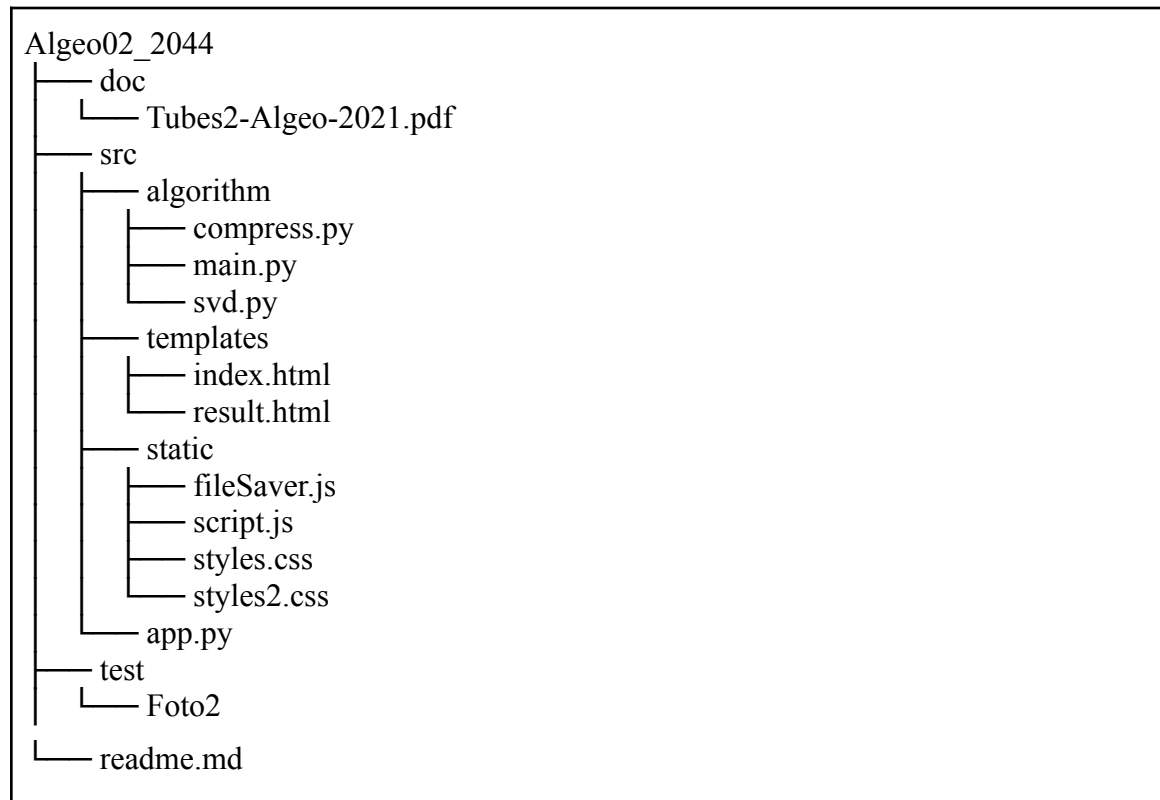
Implementasi

Bahasa Pemrograman: Python

Framework:

- a) Frontend: HTML, CSS, JS
- b) Backend: Flask

Struktur dari program kami adalah sebagai berikut:



A. Penjelasan Tech Stack

Di tugas besar kali ini, kami menggunakan Tech Stack berupa *Vanilla Js* untuk *Frontend*, dan *Flask* untuk *Backend*.

1. Frontend

Untuk bagian Frontend, kami membaginya menjadi 2 file yaitu `index.html` dan `result.html`, dikarenakan kami ingin membuat 2 *page* yang berbeda untuk *landing page* dan *page* yang sudah berhasil mengompres.

2. Backend

Untuk bagian Backend, kami menggunakan Flask karena mudah dimengerti dan mudah untuk diintegrasikan dengan algoritma SVD yang menggunakan *python*.

B. Algoritma Kompresi

1. SVD Module

- a. function randomUnitVector
Fungsi randomUnitVector menerima satu nilai N, lalu menghasilkan array yang berisi N buah vektor satuan random.
- b. function svd1D
Fungsi svd1D menerima satu buah matriks dan mengembalikan matriks V dengan menggunakan *greedy method*.
- c. function svdGreedyMethod
Fungsi svdGreedyMethod menerima matriks dan nilai K, lalu memproses svd dengan menggunakan svd1D dan mengembalikan 3 buah matriks, yaitu matriks U, S, dan VT.
- d. function eigenSimultaneousPowerIteration
Fungsi eigenSimultaneousPowerIteration menerima sebuah matriks dan nilai K, lalu mencari nilai eigen dan eigen vektornya dengan metode QR yang dilakukan iterasi sebanyak yang diinginkan. Semakin besar iterasi yang dipakai maka akan semakin akurat hasil yang diberikan.
- e. function svdPowerIteration
Fungsi svdPowerIteration mengimplementasi nilai eigen dan eigen vector dari eigenSimultaneousPowerIteration untuk mencari matriks U, S, dan VT.

2. Compress Module

- a. function openImage
Fungsi openImage berperan untuk membuka file yang diterima dari Backend lalu memprosesnya menjadi array dan mengembalikan array yang berisi matriks Red, Green, Blue, beserta gambar aslinya yang diproses library Image.
- b. function openImageAlpha
Fungsi openImage berperan untuk membuka file yang diterima dari Backend lalu memprosesnya menjadi array dan mengembalikan array yang berisi matriks Red, Green, Blue, Alpha beserta gambar aslinya yang diproses library Image. Fungsi ini untuk memproses gambar yang memiliki transparansi.
- c. function compress
Fungsi compress menerima matriks lalu mengubahnya menjadi matriks rank K melalui proses SVD.
- d. function compressedFileSize
Fungsi compressedFileSize menerima file image dan persentase kompresi untuk memperkirakan besarnya ukuran file yang nantinya akan diproses di main untuk memperkirakan banyaknya nilai singular.

3. Main Module

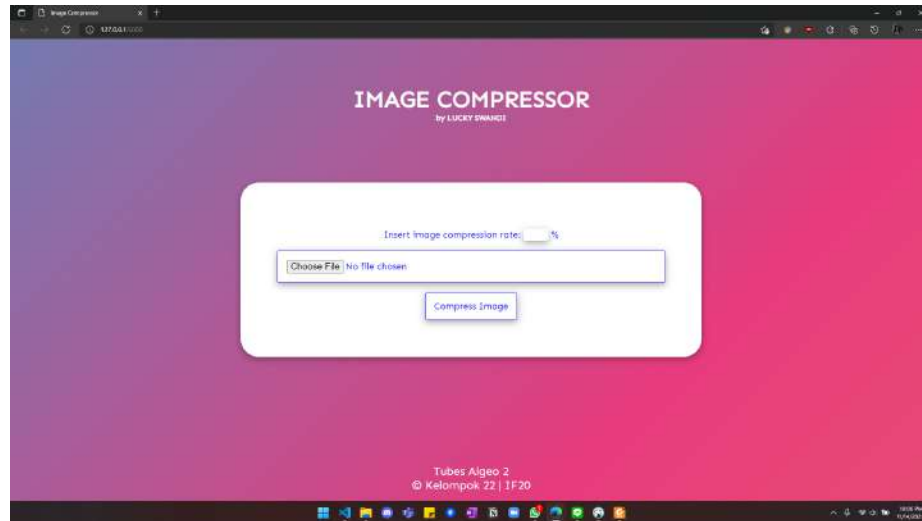
Modul main memiliki satu fungsi, yaitu fungsi main yang menerima file berupa image dan persentase kompresi yang diinginkan. Di modul ini menggabungkan semua modul yang telah dibuat untuk menghasilkan gambar yang terkompresi, waktu kompresi, dan perbedaan pixel setelah dikompresi.

BAB 4

Experimen

A. Interface

a) Tampilan website sebelum *toggle*







b) Tampilan website setelah *toggle*





B. Studi Kasus




a) binjai.jpeg

Gambar asli		Ukuran file: 68.3 kb Ukuran Matriks Gambar: 385 x 576 Total Singular Values: 385 Kasus Gambar: JPEG
Compress 15%		Singular values yang digunakan: 34 Waktu: 0.24 Detik Size: 37.9 kb Perubahan Pixels: 71.7%
Compress 20%		Singular values yang digunakan: 46 Waktu: 0.23 Detik Size: 39.5 kb Perubahan Pixels: 70.93%
Compress 25%		Singular values yang digunakan: 57 Waktu: 0.21 Size: 40.6 kb Perubahan Pixels: 70%

Compress 50%		Singular values yang digunakan: 115 Waktu: 0.37 Detik Size: 42.5 kb Perubahan Pixels: 62.24
Compress 60%		Singular values yang digunakan: 138 Waktu: 0.52 Detik Size: 42.9 kb Perubahan Pixels: 56.76%
Compress 70%		Singular values yang digunakan: 161 Waktu: 0.5 Size: 42.8 kb Perubahan Pixels: 49.34%

b) lena.png

Gambar asli		Ukuran file: 463 kb Ukuran Matriks Gambar: 512 x 512 Total Singular Values: 512 Kasus Gambar: PNG Non-Transparent
Compress 20%		Singular values yang digunakan: 25 Waktu: 0.22 Detik Size: 427 kb Perubahan Pixels: 69.78 %



Compress 40%		Singular values yang digunakan: 102 Waktu: 0.42 Detik Size: 478 kb Perubahan Pixels: 66.99%
Compress 50%		Singular values yang digunakan: 127 Waktu: 0.37 Detik Size: 493 kb Perubahan Pixels: 65.45%
Compress 60%		Singular values yang digunakan: 153 Waktu: 0.43 Detik Size: 477 kb Perubahan Pixels: 63.81%



Compress
70%





Singular values yang
digunakan: 179
Waktu: 0.53 Detik
Size: **440 kb**
Perubahan Pixels: 61.97%




c) angel.jfif




Gambar asli		Ukuran file: 68.9 kb Ukuran Matriks Gambar: 1199 x 1056 Total Singular Values: 1056 Kasus Gambar: JFIF
Compress 5%		Singular values yang digunakan: 28 Waktu: 0.79 Detik Size: 81.6 kb Perubahan Pixels: 42.19%

Compress 10%		Singular values yang digunakan: 56 Waktu: 1.08 Detik Size: 1.1 mb Perubahan Pixels: 42.32%
Compress 20%		Singular values yang digunakan: 112 Waktu: 1.25 Detik Size: 1.29 mb Perubahan Pixels: 41.17%




<p>Compress 40%</p>		<p>Singular values yang digunakan: 224 Waktu: 1.64 Detik Size: 1.39 mb Perubahan Pixels: 40.07%</p>
<p>Compress 80%</p>		<p>Singular values yang digunakan: 448 Waktu: 2.9 Detik Size: 1.25 mb Perubahan Pixels: 37.51%</p>



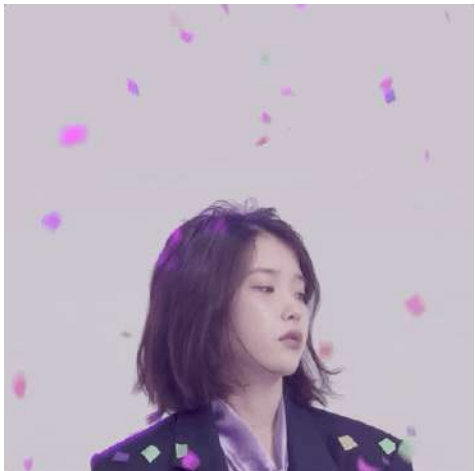
d) google.png

Gambar asli		<p>Ukuran file: 52.4 kb</p> <p>Ukuran Matriks Gambar: 1024 x 1024</p> <p>Total Singular Values: Kasus Gambar: PNG</p> <p>Transparent</p>
Compress 5%		<p>Singular values yang digunakan: 25</p> <p>Waktu: 0.53 Detik</p> <p>Size: 0.44 kb</p> <p>Perubahan Pixels: 53.63%</p>
Compress 10%		<p>Singular values yang digunakan: 51</p> <p>Waktu: 0.62 Detik</p> <p>Size: 0.52 kb</p> <p>Perubahan Pixels: 48.95%</p>

Compress 20%		Singular values yang digunakan: 102 Waktu: 0.95 Detik Size: 0.62 kb Perubahan Pixels: 45.07%
Compress 40%		Singular values yang digunakan: 204 Waktu: 1.36 Detik Size: 0.72 kb Perubahan Pixels: 41.99%
Compress 80%		Singular values yang digunakan: 409 Waktu: 2.44 Detik Size: 0.68 kb Perubahan Pixels: 38.7%

e) image5.png

Gambar asli		<p>Ukuran file: 1.74 mb</p> <p>Ukuran Matriks Gambar: 480 x 480</p> <p>Total Singular Values: 480</p> <p>Kasus Gambar: GIF</p>
Compress 5%		<p>Singular values yang digunakan: 11</p> <p>Waktu: 0.16 Detik</p> <p>Size: 0.17 kb</p> <p>Perubahan Pixels: 61.73%</p>
Compress 10%		<p>Singular values yang digunakan: 23</p> <p>Waktu: 0.17 Detik</p> <p>Size: 0.19 kb</p> <p>Perubahan Pixels: 56.99%</p>

Compress 20%		Singular values yang digunakan: 47 Waktu: 0.17 Detik Size: 0.24 kb Perubahan Pixels: 52.1%
Compress 40%		Singular values yang digunakan: 95 Waktu: 0.26 Detik Size: 0.28 kb Perubahan Pixels: 49.11%
Compress 80%		Singular values yang digunakan: 191 Waktu: 0.46 Detik Size: 0.38 kb Perubahan Pixels: 42.43%

BAB 5

Kesimpulan, Saran, Refleksi

A. Kesimpulan

Singular value decomposition merupakan metode dekomposisi matriks untuk mereduksi sebuah matriks menjadi faktor-faktor penyusunnya agar perhitungan matriks berikutnya menjadi lebih sederhana. Nilai eigen dan vektor eigen diperlukan dalam proses menentukan SVD. SVD dapat diaplikasikan pada kompresi gambar dan video, pengolahan citra, machine learning, computer vision, digital watermarking, dan lain-lain.

Pada tugas besar kedua mata kuliah IF2123 Aljabar Linear dan Geometri ini, telah berhasil dibuat web yang berfungsi untuk mengkompres suatu gambar dengan pengimplementasian nilai eigen, nilai vektor, dan singular value decomposition.

Situs web yang kelompok kami buat menerima input gambar dan juga rasio kompresi yang diinginkan. Kemudian, output yang akan ditampilkan oleh situs web adalah hasil gambar yang telah dikompres, persentase pixel gambar, dan waktu kompresi gambar. Hasil gambar yang telah dikompres juga bisa diunduh.

Jadi, tugas besar kedua IF2123 Aljabar Linear dan Geometri semester 1 Tahun 2021/2022 ini membuat kelompok memahami jalan kerja kompresi suatu gambar menggunakan pendekatan *singular value decomposition*, salah satu materi pembelajaran mata kuliah ini.

B. Saran

Adapun saran dari kelompok untuk pihak yang akan melakukan atau membuat hal yang serupa adalah:

- a) Meluangkan waktu lebih untuk mengeksplor dan mengimprove algoritma *singular value decomposition* agar proses ketika mengkompres gambar lebih cepat.
- b) Situs web dibuat dengan membuat source code frontend dan juga backend. Kelompok menyarankan agar pihak terkait melakukan eksplorasi lebih terkait frontend agar dapat menunjukkan gambar sebelum dan sesudah kompres.

C. Refleksi

Dalam pembuatan tugas besar kedua mata kuliah IF2123 Aljabar Linear dan Geometri, kelompok harus mengeksplorasi mengenai program kompresi gambar dengan pendekatan *singular value decomposition*. Terdapat kendala saat pembuatan algoritma SVD ini yaitu jalannya program sangat lama sehingga untuk mengkompres satu gambar bisa memakan waktu ber menit-menit. Akhirnya, kelompok mencoba mencari alternatif lain untuk mengimplementasikan SVD ini. Dengan menggunakan power iteration untuk melakukan pendekatan SVD terhadap matriks gambar, kelompok dapat membuat program yang dapat mengkompres suatu gambar dengan waktu yang lebih singkat. Dalam proses pembuatan web ini, kelompok juga

melakukan eksplorasi untuk membuat web yang menarik, user friendly, dan dapat berfungsi dengan baik.

Referensi

Informatika.stei.itb.ac.id/~rinaldi.munir. (2021). Algeo#01 Review Matriks. Diakses pada 14 November 2021, dari

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-01-Review-Matriks.pdf>

http://mlwiki.org/index.php/Power_Iteration

<https://jeremykun.com/2016/04/18/singular-value-decomposition-part-1-perspectives-on-linear-algebra/>

<https://jeremykun.com/2016/05/16/singular-value-decomposition-part-2-theorem-proof-algorithm/>

<https://flask.palletsprojects.com/en/2.0.x/>