

Laporan Tugas Besar II

IF2123 Aljabar Linier dan Geometri

Oleh:

Kelompok ALGE-DEAD

Anggota:

Vieri Mansyl (13520092)

Vincent Prasetya Atmadja (13520099)

Steven (13520131)



SEKOLAH TINGGI ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021/2022

DAFTAR ISI

DAFTAR ISI	2
BAB I DESKRIPSI MASALAH.....	1
1.1. Latar Belakang.....	1
1.2. Spesifikasi Tugas	2
1.3. Spesifikasi Program	2
BAB II TEORI SINGKAT.....	3
2.1 Perkalian matriks	3
2.2 Nilai eigen dan vektor eigen	3
2.3 Matriks SVD	4
BAB III IMPLEMENTASI.....	5
3.1. Eigen.py	5
3.2. Compress.py.....	5
3.3. App.py.....	6
3.4. Index.html	6
3.5. Index.css.....	6
BAB IV EKSPERIMEN.....	7
4.1. Gambar dengan ekstensi .jpg	7
4.2. Gambar dengan ekstensi .jpeg	9
4.3. Gambar dengan ekstensi .png	11
BAB V KESIMPULAN	13
5.1 Simpulan	13
5.2 Saran	13
5.3 Refleksi	13
REFERENSI.....	14

BAB I

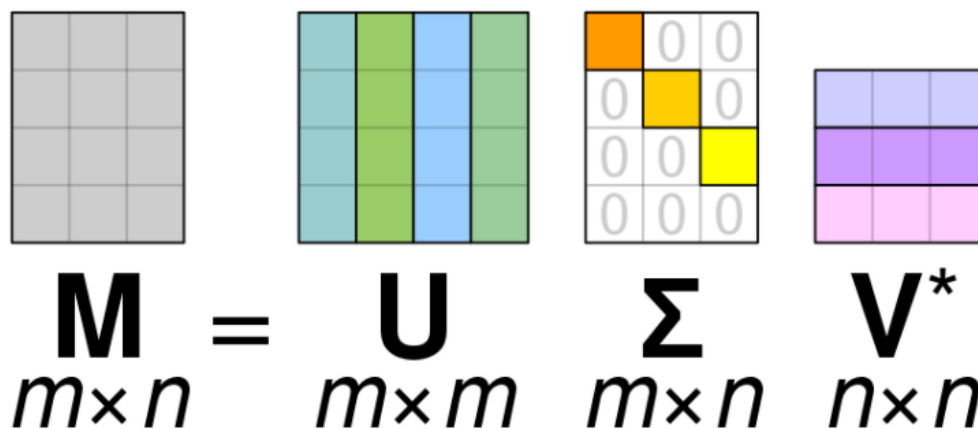
DESKRIPSI MASALAH

1.1. Latar Belakang

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.

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 .



$$\begin{matrix}
 \text{4x4} & & \text{4x4} & & \text{4x4} & & \text{4x4} \\
 \mathbf{M} & = & \mathbf{U} & & \mathbf{\Sigma} & & \mathbf{V}^* \\
 m \times n & & m \times m & & m \times n & & n \times n
 \end{matrix}$$

Gambar 1.1.1. Ilustrasi Dekomposisi menggunakan metode SVD

Pada Tubes 2 ini, hendak dibuat website kompresi gambar sederhana dengan menggunakan algoritma SVD.

1.2. Spesifikasi Tugas

Buatlah program kompresi gambar dengan memanfaatkan algoritma SVD dalam bentuk website lokal sederhana.

1.3. Spesifikasi Program

- 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.
- 11 **Dilarang** menggunakan library perhitungan SVD dan library pengolahan eigen yang sudah jadi.

BAB II

TEORI SINGKAT

2.1 Perkalian matriks

Dalam aljabar linier, perkalian matriks merupakan operasi perkalian yang menghasilkan matriks baru dari dua matriks. Apabila matriks A dikalikan dengan matriks B, maka kolom dari matriks A harus berjumlah sama dengan baris dari matriks B. Hasil perkalian dari kedua matriks (misalkan AB) memiliki jumlah baris setara dengan jumlah baris matriks A dan jumlah kolom setara dengan jumlah kolom matriks B, sehingga pernyataan ini dapat ditulis sebagai berikut.

$$AB_{m \times n} = A_{m \times k} \cdot B_{k \times n}$$

persamaan 2.1.1 perkalian 2 matriks

2.2 Nilai eigen dan vektor eigen

Jika A adalah matriks persegi yang berukuran $n \times n$, maka vektor tidak-nol \mathbf{x} di \mathbb{R}^n disebut vektor eigen dari A jika $A\mathbf{x}$ sama dengan perkalian suatu skalar λ dengan \mathbf{x} , yaitu

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

dengan λ merupakan nilai eigen dari A dan \mathbf{x} merupakan vektor eigen yang berkoresponden dengan λ . (kata “eigen” berasal dari Bahasa Jerman memiliki arti “asli” atau “karakteristik”).

Untuk menghitung nilai eigen dari suatu matriks A berukuran $n \times n$, dapat dilakukan langkah-langkah berikut.

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

persamaan 2.2.1. sistem persamaan homogen matriks $(A - I\lambda)$

Agar $(A - I\lambda)\mathbf{x} = 0$ memiliki solusi tidak-nol selain $\mathbf{x} = 0$, maka

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

Hasil dari persamaan tersebut akan berupa akar-akar λ yang dinamakan akar-akar karakteristik atau nilai-nilai eigen.

Dengan memasukkan setiap nilai λ pada persamaan 2.2.1, maka solusi dari SPL tersebut akan membentuk vektor-vektor eigen.

2.3 Matriks SVD

Suatu matriks A yang berukuran $m \times n$ dapat difaktorkan / didekomposisi menjadi matriks orthogonal serta matriks diagonal dengan menggunakan metode Singular Value Decomposition (SVD), sehingga matriks A dapat ditulis sebagai berikut.

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

dimana :

A = matriks berukuran $m \times n$

U = matriks orthogonal berukuran $m \times m$ (disebut sebagai matriks singular kiri)

Σ = matriks diagonal berukuran $m \times n$ (elemen diagonalnya merupakan nilai-nilai singular dari matriks A)

V^T = matriks orthogonal berukuran $n \times n$ (disebut sebagai matriks singular kanan)

Untuk membentuk matriks SVD dari matriks A , dilakukan langkah-langkah berikut.

1. Matriks V merupakan himpunan vektor eigen di mana V mendiagonalisasi matriks $A^T A$.
2. Dari matriks $A^T A$, didapat nilai eigen $\lambda_1, \lambda_2, \dots, \lambda_n$ sedemikian sehingga nilai-nilai singular $\sigma_1 = \sqrt{\lambda_1}, \sigma_2 = \sqrt{\lambda_2}, \dots, \sigma_n = \sqrt{\lambda_n}$ merupakan elemen dari matriks Σ . (catatan : $\lambda_1, \lambda_2, \dots, \lambda_n$ merupakan nilai eigen tak-nol yang terurut mengecil.)
3. Vektor kolom V terurut sehingga $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$.
4. Vektor $u_i = \frac{1}{\sigma_i} A v_i$ ($i = 1, 2, \dots, n$)
5. Himpunan dari vektor-vektor u_i akan membentuk matriks singular U .

BAB III

IMPLEMENTASI

Website kompresi gambar dibuat dengan menggunakan bahasa pemrograman Python, HTML, dan CSS , serta mengimplementasikan berbagai fungsionalitas ke dalam 3 buah file, yaitu file `eigen.py` yang mengembalikan nilai-nilai serta vektor eigen dari suatu matriks , file `compress.py` yang merupakan realisasi kompresi gambar, dan file `app.py` sebagai penghubung antara *backend* dengan *frontend*. File HTML berupa `Index.html` untuk membentuk main page website serta file CSS berupa `Index.css` untuk memperindah main page website.

3.1. Eigen.py

File ini berisikan 4 fungsi sebagai bentuk fungsionalitas eigen, yaitu :

1. `rounding(val)` berfungsi melakukan pembulatan dengan toleransi sebesar 10^{-9} terhadap suatu nilai `val` apabila nilai `val` mendekati dengan nilai bulat terdekatnya dan mengembalikan nilai dari `val` tersebut.
2. `convEigSig(rawroot , rawvec)` mengembaliakn list yang berisi nilai eigen , nilai singular , dan vektor eigen.
3. `findEigen(mat , maxiter)` berfungsi menghitung nilai eigen dengan menggunakan metode dekomposisi QR dan mengembalikan nilai eigen dan vektor eigen.
4. `createSigmaMat(sigma , nrow , ncol)` berfungsi membentuk matriks Σ berdasarkan nilai-nilai singular yang didapat.

3.2. Compress.py

File ini berisikan 5 buah fungsi yang merupakan bagian dari pemrosesan utama compression image dan svd

1. `Compress(image , compression_rate)` berfungsi melakukan compression image. Fungsi ini memiliki 2 paramater, `image` adalah path ke foto yang akan decompress di folder, dan `compression rate` adalah `compression_rate` yang diinginkan.
2. `Pixdiff(nrow , ncol , k)` berfungsi untuk menghitung `pixel_difference` foto setelah dan sebelum decompress. Parameter yang diterima adalah `nrow`, ukuran lebar foto, `ncol`, adalah ukuran Panjang foto, dan `k` adalah banyaknya singular value yang digunakan
3. `findK(cr , singVal)` berfungsi untuk menghitung banyaknya singular value yang akan digunakan berdasarkan compression rate. Parameter yang diterima adalah `cr`, compression rate dalam persen dan `singVal` adalah banyak singular Value keseluruhan dari foto
4. `process(mat , cr , maxiter)` berfungsi untuk melakukan svd dari masing-masing channel. Parameter yang diterima adalah `mat`, yaitu matriks yang akan dicari svd-nya dan

decompress, cr adalah compression rate, dan maxiter adalah iterasi maksimal yang akan digunakan untuk looping dalam mencari eigen. Maxiter adalah nrow atau ncol yang lebih besar.

5. `multiplyMat(mu , sig , mv)` fungsi ini berguna untuk melakukan perkalian 3 buah matriks berurutan sesuai dengan input, yaitu $\mu * \sigma * mv$.

3.3. App.py

File ini merupakan file yang digunakan untuk melakukan linking antara front-end dan back-end. Metode linking yang digunakan adalah metode POST and GET. Dengan menggunakan kedua metode tersebut, terbentuklah linking antara front-end dengan back-end. Singkatnya linking terjadi dengan cara menerima inputan user pada website, lalu mengolahnya dengan fungsi-fungsi yang telah kami buat kemudian mengoutputkan hasilnya.

3.4. Index.html

File ini berisi markup-language yang digunakan sebagai main page pada website. Di dalam page website ini tersedia fitur upload photo, input compression rate, dan fitur download. Tidak hanya itu, page website ini juga menampilkan foto before after dari suatu foto yang dikompres.

3.5. Index.css

File ini berfungsi untuk memperindah tampilan dari main page (index.html) baik dari segi ukuran font, alignment, dan lain-lain.

BAB IV

EKSPERIMEN

Berikut merupakan tes uji terhadap gambar-gambar berdasarkan perbedaan extension file untuk mengukur kebenaran dari program yang telah dibuat. Ekstensi file yang akan di uji coba ialah .jpg , .jpeg , .png, dengan resolusi yang bervariasi.

Note: 0 tidak terkompres sama sekali, 100 terkompres maksimal

4.1. Gambar dengan ekstensi .jpg

Compression rate : 100%

Sebelum kompresi :



Setelah kompresi :



Compression rate : 80%
Sebelum kompresi :



Setelah kompresi :



Compression rate : 50%
Sebelum kompresi :



Setelah kompresi :



Compression rate : 0%
Sebelum kompresi :



Setelah kompresi :

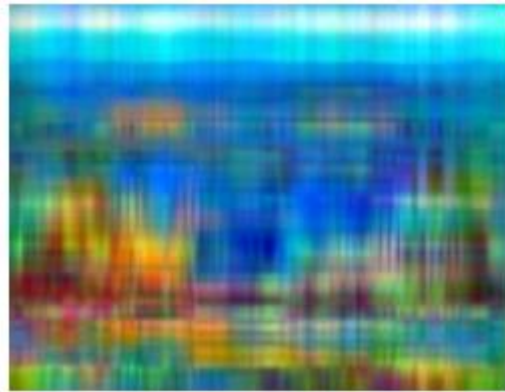


4.2. Gambar dengan ekstensi .jpeg

Compression rate : 100%
Sebelum kompresi :



Setelah kompresi :



Compression rate : 80%
Sebelum kompresi :



Setelah kompresi :



Compression rate : 50%
Sebelum kompresi :



Setelah kompresi :



Compression rate : 0%
Sebelum kompresi :



Setelah kompresi :



4.3. Gambar dengan ekstensi .png

Compression rate : 100%

Sebelum kompresi :



Setelah kompresi :



Compression rate : 80%

Sebelum kompresi :



Setelah kompresi :



Compression rate : 50%
Sebelum kompresi :



Setelah kompresi :



Compression rate : 0%
Sebelum kompresi :



Setelah kompresi :



BAB V

SIMPULAN, SARAN, REFLEKSI

5.1 Simpulan

Metode dekomposisi SVD dapat digunakan untuk melakukan kompresi gambar, dengan ukuran matriks berdasarkan besar resolusi dari gambar. Dengan menggunakan metode dekomposisi QR dapat dicari nilai-nilai eigen yang akan dipakai dalam dekomposisi SVD.

Website yang kami kerjakan dibuat menggunakan Flask, HTML, dan CSS. Flask digunakan sebagai linker antara *frontend* dengan *backend* (dengan menggunakan POST and GET), HTML digunakan sebagai penampil main page, dan CSS digunakan sebagai perias dari file HTML.

5.2 Saran

Untuk penelitian selanjutnya, program dapat dikembangkan pada aspek efisiensi waktu. Selain itu, apabila ingin mencari nilai eigen dengan menggunakan program gauss pada tubes sebelumnya, program dapat di-*develop* untuk mencegah kompleksitas waktu yang terlalu besar.

5.3 Refleksi

Dari tugas besar ini, kami telah berhasil memahami algoritma dari metode SVD serta cara kerja kompresi suatu gambar dengan menggunakan metode SVD. Dikarenakan deadline tubes ini pula yang cukup singkat, kami telah berhasil membuat program kompresi gambar menggunakan metode SVD serta dapat menampilkannya pada website sesuai spesifikasi tubes dengan menyelesaikannya dengan tepat waktu.

REFERENSI

1. <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2021-2022/>
2. <http://timbaumann.info/svd-image-compression-demo/>
3. <https://www.andreinc.net/2021/01/25/computing-eigenvalues-and-eigenvectors-using-qr-decomposition>
4. <https://numpy.org/doc/>