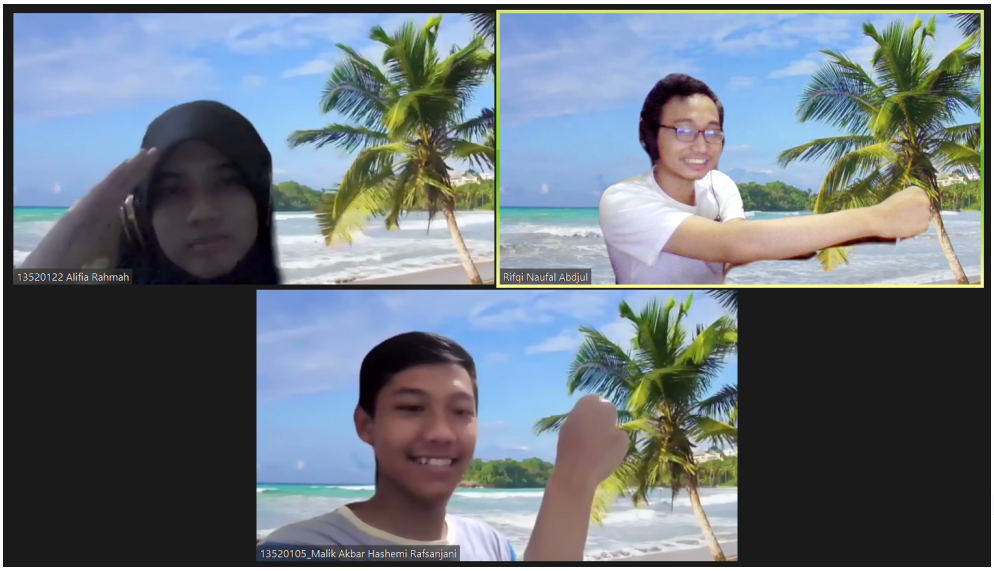
Tugas Besar 2
IF 2123 Aljabar Linier dan Geometri



Oleh:

Rifqi Naufal Abdjul	13520062
Malik Akbar Hashemi Rafsanjani	13520105
Alifia Rahmah	13520122

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021

DAFTAR ISI

BAB 1
Deskripsi Masalah

I. Tujuan

1. Membuat aplikasi berbasis web untuk melakukan kompresi gambar.

II. Spesifikasi

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. 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^TA. Matriks ini menyimpan informasi yang penting terkait kolom-kolom matriks awal, dengan informasi terpenting disimpan dalam baris pertama.

Pada Tugas besar 2 ini, diberikan tantangan untuk membuat website kompresi gambar sederhana dengan menggunakan algoritma SVD.

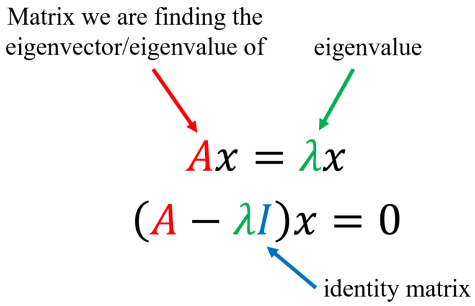
BAB 2
Teori Singkat

Perkalian Matriks

Pada perkalian antar dua matriks memiliki metode, diperlukan syarat banyak kolom pada matriks pertama dan banyak baris pada matriks kedua yang sama. Setiap elemen pada baris m dan kolom n di matriks hasil perkalian didapatkan dari penjumlahan semua perkalian tiap elemen pada kolom m di matriks pertama dan baris n pada baris kedua.
Secara formal, pada matriks $C_{m \times n} = A_{m \times r} B_{r \times n}$, Misal $A = [a_{ij}]$ dan $B = [b_{ij}]$, maka $C = A \times B = [c_{ij}]$, $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{in}b_{nj}$.

Nilai Eigen

Nilai eigen merupakan nilai karakteristik dari suatu matriks berukuran n x n (matriks persegi). Jika A merupakan matriks persegi, maka vektor tak nol x dinamakan vektor eigen dari A jika Ax merupakan kelipatan skalar dari x, yang mana skalar tersebut dinamakan nilai eigen. Hal tersebut dapat dinyatakan dalam persamaan sebagai berikut.



Vektor Eigen

Hampir sama dengan nilai eigen, vektor eigen merupakan vektor karakteristik dari suatu matriks berukuran n x n. Setiap vektor eigen berkorespondensi dengan sebuah nilai eigen. Suatu vektor eigen haruslah merupakan vektor tak nol, yang mana jika vektor eigen merupakan vektor nol, maka terdapat tak berhingga nilai eigen yang berkorespondensi.

Dekomposisi SVD

Dekomposisi singular value (SVD) merupakan faktorisasi dari matriks real atau matriks complex M yang berukuran n x m menjadi 3 value yaitu U, Σ, V* dimana U adalah matriks singular kiri dengan ukuran n x n, Σ adalah matriks persegi panjang yang berisikan nilai singular dari matriks M, dan V merupakan matriks singular kanan dengan ukuran m x m.

Matriks U merupakan matriks ortonormal ($A^{-1} = A^T$) yang dibentuk dari ruang basis vektor eigen dari matriks $M.M^T$ yang telah dinormalisasi. Matriks Σ merupakan matriks diagonal yang berbentuk persegi panjang mengikuti ukuran matriks M. Matriks V merupakan matriks ortonormal yang dibentuk dari ruang basis vektor eigen dari matriks M^TM yang telah dinormalisasi.

Dekomposisi QR

Dekomposisi QR merupakan faktorisasi matriks misal A, menjadi 2 matriks Q dan R dimana Q merupakan matriks orthogonal ($Q^TQ = QQ^T = I$) dan R merupakan matriks segitiga atas. Dekomposisi QR biasa digunakan untuk menghitung eigenvalue dan eigenvector dikarenakan sifat dekomposisi QR yang menjaga nilai eigennya.

Power Method

Power method merupakan metode yang digunakan untuk meng-aproksimasi eigenvalue dominan (yang mempunyai nilai mutlak yang paling besar), dan pasangan eigenvector nya. Power method dilakukan dengan cara mengalikan dan normalisasi vektor tebakan dengan matriks awal. Sehingga, vektor akan konvergen ke arah eigenvector yang paling dominan dari matriks awal. Secara matematis, power method dirumuskan seperti berikut:

$$u_i = \frac{Au_{i-1}}{||Au_{i-1}||}$$

Dimana,

u_i = aproksimasi eigenvector ke-i

i = jumlah iterasi, dan saat i = 0, u merupakan vektor asal

A = Matriks awal

Orthogonal Iteration

Merupakan generalisasi dari power method, dimana metode ini akan langsung menghasilkan aproksimasi seluruh eigenvector dan eigenvalue dengan cara mengalikan banyak vektor yang diletakkan di matriks Q terhadap A. Untuk normalisasi ruang vektor, dapat digunakan dekomposisi QR agar menghasilkan matriks Q dimana $Q^TQ = QQ^T = I$. Ketika hasil telah konvergen, kolom dari matriks Q merupakan eigenvector dan diagonal dari matriks R merupakan eigenvalue yang berpasangan dengan sesuai dengan urutannya dengan eigenvector di matriks Q

BAB 3
Implementasi

Tech Stack

Back-End

Pada bagian backend, kami menggunakan bahasa Python dengan *framework* Flask. Kami memilih bahasa Python karena bahasa ini juga kami gunakan untuk mengkalkulasi kompresi gambar sehingga memudahkan integrasi dari aplikasi website dengan algoritma kompresi. Kami memilih *framework* Flask karena *framework* ini relatif lebih sederhana untuk pengimplementasian aplikasi website, terlebih aplikasi website kami dapat dibilang cukup sederhana pula. Namun, *framework* Flask tetaplah mendukung *routing* dan *dynamic values*.

Front-End

Pada bagian frontend, kami menggunakan bahasa JavaScript tanpa penambahan library ataupun *framework* (Vanilla JavaScript) serta menggunakan framework dengan tambahan beberapa *plain* CSS. Di samping kecepatan dari Vanilla JavaScript, kami memilih hal tersebut karena integrasi dengan *template* dari Flask yang jauh lebih mudah, tanpa mengurangi kemampuan meng-*handle dynamic values*. Hampir sama dengan alasan pemilihan Vanilla JavaScript, kami memilih menggunakan *plain* CSS karena kemudahan integrasi, yang mana kami tidak perlu melakukan instalasi berbagai modules, tanpa mengurangi kemampuan *styling* ke tampilan website. Penggunaan Bootstrap juga dilakukan untuk memudahkan penempatan elemen-elemen dalam halaman website.

Struktur Directory

```
├── Pipfile
├── Pipfile.lock
├── README.md
├── doc
│   ├── Laporan Algeo02-20062.pdf
│   └── Spek Tubes Algeo 2 2021.pdf
├── requirements.txt
├── src
│   ├── flask_app.py
│   ├── src
│   │   ├── compressor.py
│   │   └── forms.py
│   ├── static
│   │   ├── assets
│   │   │   ├── favicon.ico
│   │   │   ├── layout-web-with-images.png
```

```
│   │   │   ├── layout-web.png
│   │   │   ├── loading.gif
│   │   │   └── palm-tree-emoji.png
│   │   ├── downloads
│   │   ├── script.js
│   │   ├── style.css
│   │   └── uploads
│   └── templates
│       ├── index.html
│       └── upload.html
└── test
    ├── test_compression.png
    ├── test_compression2.jpeg
    └── web_interface.png
```

Pada root folder terdapat file README.md yang berisi penjelasan program, file Pipfile, file Pipfile.lock, serta folder-folder lain, seperti: folder src, folder test, dan folder doc. Pada folder src terdapat file flask_app.py berisi program utama flask app, serta folder folder lain, seperti, src, static, dan templates. Pada folder src berisi module python program yaitu forms.py untuk mengambil data dari form html menggunakan library wtform dan compressor.py untuk mengolah gambar yang telah di-*upload* oleh pengguna. Folder static berisi file script.js untuk program JavaScript *frontend*, dan style.css untuk *styling* pada *frontend*, serta folder uploads untuk menyimpan gambar unggahan pengguna dan folder downloads untuk menyimpan hasil kompresi, serta folder assets untuk menyimpan logo icon website dan gambar animasi ketika *loading* kompresi. Pada folder templates terdapat file index.html sebagai root dari file html yang akan ditampilkan ke pengguna dan file upload.html sebagai file html tambahan ke index.html ketika pengguna telah mengunggah gambar. Pada folder test berisi gambar masukan untuk pengujian, sementara folder doc berisi laporan dengan format PDF.

Algoritma

Orthogonal Iteration

Pada bagian ini yang digunakan untuk mencari eigenvalue dan eigenvector, kami menggunakan modul numpy untuk melakukan pengalian matriks dan normalisasi matriks (QR decomposition). Pada bagian ini juga kami mengimplementasi agar dapat mengatur jumlah eigenvector dan eigenvalue yang diinginkan sesuai dengan masukan k. Secara garis besar, gambaran fungsinya adalah seperti berikut,

1. Inisialisasi Q_0 dengan nilai random berukuran n x k, dengan n merupakan jumlah baris matriks input dan k merupakan jumlah pasangan eigen yang diinginkan
2. Normalisasi Q_0 menggunakan dekomposisi QR
3. Melakukan iterasi dengan langkah
    a. Perkalian dot A dengan Q_i menghasilkan misal matriks Z_i
    b. Dekomposisi matriks Z_i menghasilkan Q_{i+1} dan R_{i+1}
4. Mengembalikan nilai diagonal R terakhir dan matriks Q terakhir

Dekomposisi SVD

Pada fungsi ini kami menggabungkannya sekaligus dengan kompresi dikarenakan fungsi untuk mendapatkan eigenvector dan eigenvaluenya hanya yang dibutuhkan. Untuk memproses matriks dengan ukuran n x m kami mempunyai 2 kasus yaitu saat $n < m$ dan $n \geq m$ untuk mempercepat proses dan tidak mengecek seluruh

Untuk kasus $n < m$, kami mencari matriks U dan Σ dan selanjutnya mencari matriks V^T menggunakan metode invers dengan matriks awalnya.

$$A = U\Sigma V^T$$
$$V^T = \Sigma^{-1} U^T A$$

Sedangkan untuk kasus $n \geq m$ kami mencari matriks Σ dan V^T dan selanjutnya mencari matriks U menggunakan metode invers dengan matriks awalnya.

$$A = U\Sigma V^T$$
$$U = \Sigma^{-1} V A$$

Read and Write Gambar

Pada proses pembacaan gambar dari input, kami menggunakan PIL untuk memprosesnya dan melakukan splitting berdasarkan jumlah channel dari gambar, apakah 3 channel (RGB untuk .jpg dan sejenisnya) atau 4 channel (RGBA untuk .png dan sejenisnya). Sebelum dilakukan proses kami mengubah tipe data dari Uint8 (unsigned integer 8 bit) menjadi float agar bisa diproses untuk kompresi. Setelah melewati kompresi, nilai RGB atau RGBA akan diubah kembali menjadi Uint8 (setelah dibatasi diantara 0 hingga 255).

BAB 4
Eksperimen

Gambar 4K (4000 x 3000)
Compression time : 834.36 s

File size : 3.67 MB

Compression 50% (50% pixel similarity)

File Size: 1.26 MB

Normal Image (512 x 512)
Compression time: 30.60 s

File Size : 432 KB

Compression 50% (50% pixel similarity)

File Size:382 KB

Extreme Compression (256 x 256)
Compression time: 0.09 s

File size : 35.9 KB

Compression 99% (1% pixel similarity)

File size :33.7 KB

B&W transparent (667 x 277)
Compression time: 1.6120600700378418 sec

File size: 27.6 KB

File size: 18.4 KB

BAB 5
Kesimpulan

Kesimpulan

Algoritma dekomposisi SVD merupakan salah satu algoritma yang bisa digunakan untuk mengompresi gambar. Dengan memecah matriks menjadi nilai singularnya, kita bisa memangkas beberapa nilai singular yang tidak terlalu berpengaruh dalam gambar. Walau algoritma ini kurang bagus untuk mengompresi gambar dikarenakan hanya mempengaruhi nilai data gambar, bukan cara penyimpanannya, algoritma ini cukup efektif untuk melakukan kompresi gambar.

Saran

Hasil tugas ini dapat dijadikan lebih baik dengan cara penambahan beberapa fitur lain, seperti: dapat mengompresi untuk gambar-gambar dengan ekstensi selain JPG dan PNG, pengaturan dari color depth dari gambar, mengambil gambar langsung dari URL, maupun kompresi beberapa gambar secara sekaligus.

Refleksi

a. Rifqi Naufal Abdjul (13520062)

   Saya belajar banyak teori algoritma dari tugas besar ini, mulai dari metode komputasi yang menggunakan iterasi dibandingkan menghitung langsung, membandingkan kompleksitas algoritma, dan yang terkait optimisasi algoritma untuk mempercepat proses karena metode yang naif sehingga jauh lebih lambat daripada biasanya. Saya juga belajar banyak teori matematika walau hanya sekilas tentang eigen dan jenis jenis matriks saat eksplorasi.

b. Malik Akbar Hashemi Rafsanjani (13520105)

   Saya banyak belajar pada tugas besar ini, mulai dari *web development* hingga pengolahan gambar. Saya belajar mulai dari inisialisasi *website development project*, implementasi backend, integrasi *backend*, *frontend*, dan juga modul pengolahan yang dipakai di server, serta bagaimana men-*deploy* flask app. Saya juga belajar mengenai representasi gambar pada komputer, bagaimana ekstraksi dan pengolahannya. Saya semakin sadar bahwa masih sangat banyak dan luas hal-hal yang dapat saya pelajari dan dalami.

c. Alifia Rahmah (13520122)

   Saya banyak eksplorasi mengenai dunia *web development* di tugas besar ini, terutama di sisi *front end*. Salah satu pelajaran berharga yang saya dapatkan dari tugas besar ini adalah jika kita tidak tahu bagaimana melakukan sesuatu dengan salah satu alat, ada baiknya kembali ke dasar (*back to basics*). Beberapa fungsi saya implementasikan dengan Javascript, dan beberapa *style* yang tidak bisa dipenuhi oleh Bootstrap saya impelementasikan dengan CSS murni. Saya juga belajar bagaimana mengintegrasikan Flask dengan styling di bagian front-end dengan Jinja.

Daftar Pustaka

Strang, G. (2016). *Introduction to linear algebra*. Cambridge Press.

Stephen Andrilli, David Hecker, in Elementary Linear Algebra (Fifth Edition), 2016

Power iteration Page ID 623. Diakses dari http://mlwiki.org/index.php/Power_Iteration

How to Flash Messages in Flask?. Diakses dari
https://www.askpython.com/python-modules/flask/flask-flash-method

How to upload and download files using Flask. Diakses dari
https://www.codeunderscored.com/upload-download-files-flask/

Form Validation with WTForms. Diakses dari
https://flask.palletsprojects.com/en/2.0.x/patterns/wtforms/

Bootstrap Documentation. Diakses dari
https://getbootstrap.com/docs/5.1/getting-started/introduction/