

LAPORAN TUGAS BESAR ALJABAR LINEAR DAN GEOMETRI SINGULAR VALUE DECOMPOSITION IF2123 ALJABAR LINEAR DAN GEOMETRI



Kelompok : YaSVDahlah

Disusun oleh :

Ilham Prasetyo Wibowo (13520013)

Gregorius Moses Marevson (13520052)

Muhammad Naufal Satriandana (13520068)

**PROGRAM STUDI TEKNIK INFORMATIKA
SETKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021**

BAB I

PENDAHULUAN

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. Dekomposisi matriks ini dapat dinyatakan sesuai persamaan berikut.

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.

$$A_{M \times N} = U_{M \times M} S_{M \times N} V_{N \times N}^T$$

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.



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.

BAB II

TEORI SINGKAT

Warna yang ditampilkan oleh monitor adalah campuran dari spektrum RGB yang dipancarkan dari pixel-pixel kecil di layar. Warna itu mempunyai representasi angka dari 0 (tidak berwarna) hingga 255 (pekat). Begitu pula dengan gambar, yang merupakan campuran dari warna- warna. Gambar itu bisa direpresentasikan sebagai matriks penyimpan nilai warna. Misalkan pada gambar grayscale berukuran 100 x 100 pixel, gambar itu bisa direpresentasikan dalam bentuk matriks 100 x 100 yang mengandung nilai 0 – 255.

Dalam kompresi dengan SVD, matriks diolah menjadi perkalian dari 3 sub-matriks ($U \cdot \Sigma \cdot VT$) yang kemudian akan dikompres dengan faktor k . Misalkan pada gambar 100 x 100 pixel, kita ingin mengompresnya dengan $k = 10$. Matriks itu awalnya dipecah menjadi $U = 100 \times 100$, $\Sigma = 100 \times 100$, dan $VT = 100 \times 100$. Nilai k digunakan untuk memartisi ketiga matriks itu menjadi $U = 100 \times 10$, $\Sigma = 10 \times 10$, dan $VT = 10 \times 100$. Jadi, alih – alih menyimpan 100 x 100 (10000 angka), kita mengompresnya menjadi (2100 angka).

BAB III

IMPLEMENTASI

3.1 Tech stack yang digunakan:

Website dibuat menggunakan framework Flask. Flask itu sendiri adalah sebuah web framework dengan bahasa python yang menyediakan library dan sekumpulan kode yang siap dipakai. Secara umum, Flask menggunakan Werkzeug untuk menerima request url dan Jinja2 untuk merender template yang ada pada file html.

Website ditulis dengan bahasa python, HTML, dan CSS. HTML berfungsi sebagai dasar dari struktur website. Sedangkan CSS berfungsi untuk mengubah tampilan website agar lebih user friendly. Framework CSS yang digunakan adalah Bootstrap untuk mempermudah mengubah tampilan website.

Fungsi untuk mengompres gambar dari website ditulis menggunakan bahasa python. Website akan menerima sebuah gambar, dan persentase dari r yang akan digunakan dari user. Yang dalam hal ini persentase r adalah rasio dari r/r_{\max} atau ukuran maksimum gambar. Kemudian server akan mengolah gambar dan setelah kompresi dilakukan, user bisa mengunduh gambar dari website.

3.2 Garis besar algoritma program

Dalam pengolahan gambar, digunakan library numpy, math, pillow, scipy, dan time. Program diawali dengan inputan user yang berupa gambar dan persentase r . variabel r adalah nilai singular pertama yang disimpan saat dilakukan svd kepada matriks gambar. Selanjutnya, dengan menggunakan fungsi numpy dan pillow, gambar tersebut dikonversi menjadi matriks yang berupa array numpy sebagai float.

Program melanjutkan prosesnya sesuai dengan jenis gambarnya. Jika gambar tersebut RGB, program akan memisah matriks tersebut menjadi 3 bagian, yang masing masing bagian merepresentasikan kanal merah, hijau, dan biru. Setiap bagian tersebut kemudian akan melewati proses svd untuk di faktorisasi menjadi 3 bagian. Jika gambar tersebut grayscale, program akan langsung melakukan proses svd pada gambar tersebut.

Dalam proses svd, pertama dilakukan transpose terhadap matriks input, kemudian matrix tersebut dikalikan dengan matrix awal dan disimpan nilainya (anggap ke dalam variabel ata).

Lalu dilakukan proses pengambilan eigen value dan eigen vector terhadap matrix ata ini dengan QR value decomposition. Lalu, hasil eigen value ini disortir tak naik. Kemudian dicari matriks kanan svd (V transpose) dengan cara mentranspos eigen vector. Selanjutnya, nilai sigma (matriks tengah) dicari dengan mengakarkan nilai mutlak dari eigen value dan hanya menyimpan nilai tak nol. Perlu diperhatikan, matriks sigma hanya berdimensi satu, yang dalam hal ini matriksnya hanya menyimpan nilai diagonal dari matriks sesungguhnya. Kemudian, matriks kiri (U) dicari dengan mengalikan matriks awal dengan matriks V (eigen vector) dan membaginya (tiap kolomnya) dengan sigma (yang bersesuaian).

Setelah mendapatkan nilai svd, program akan merekonstruksi ulang matriks tiap kanal (atau langsung matriks gambar tersebut untuk grayscale) dengan cara mengalikan kembali

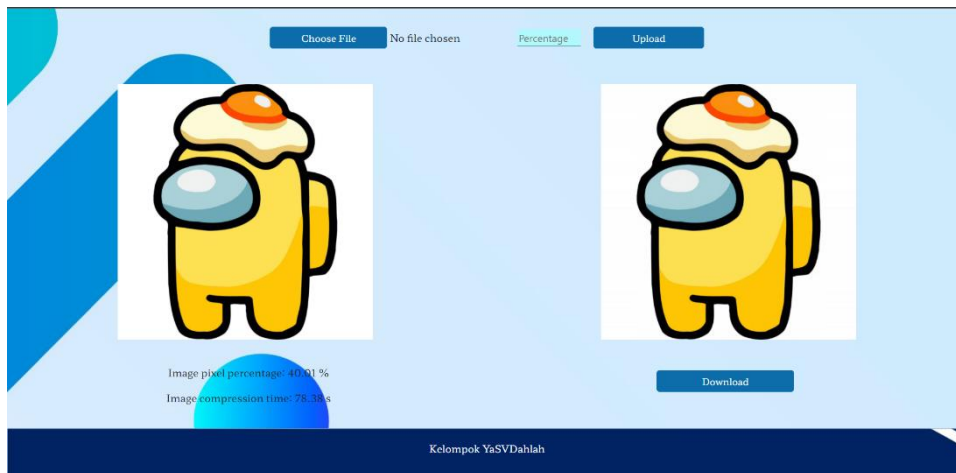
hasil svd yang telah dimodifikasi. Hasil modifikasi ini tergantung nilai r yang diterima di awal. Jika matriks utama berukuran $m \times n$ dan matrix svd berukuran $m \times k$, k , dan $k \times n$ untuk komponen U , Σ , dan V transpose secara berturut-turut, r , yang dalam hal ini $0 \leq r \leq k$, akan menggantikan k sehingga matriks svd yang sudah dimodifikasi menjadi berukuran $m \times r$, r , dan $r \times n$ secara berturut-turut. Perlu diperhatikan, matriks Σ harus di diagonalisasi terlebih dahulu agar dapat dikalikan dengan U dan V transpose.

Matriks hasil rekonstruksi kemudian akan disatukan lagi per kanal nya dan dikembalikan ke gambar. Jika gambar grayscale, tidak perlu disatukan terlebih dahulu. Jika gambar memiliki kanal alpha, yakni RGBA, caranya sama seperti RGB, namun ada tambahan satu kanal lagi, yakni alpha. Gambar ini kemudian disimpan di suatu tempat.

Setelah selesai proses, program akan menampilkan rasio kompresi, dengan rumus :
(byte penyimpanan svd modifikasi) / (byte awal) yang dalam hal ini byte dapat direpresentasikan dengan jumlah komponen matriks, atau ukuran matriks (jika matriks berukuran 800×600 maka byte dapat direpresentasikan dengan 480000).

BAB IV

EKSPERIMEN



Kompresi terhadap amogus.jpeg 1500x1500 dengan persentase $r = 20\%$

Eksperimen pertama adalah terhadap amogus.jpeg ukuran 1500x1500. Setelah melewati svd dan menyimpan 20 persen nilai pertama (20% dari 1500 adalah 300), didapatkan matriks U, Sigma, dan Vt berukuran 1500x300, 1x300, dan 300x1500 berturut turut. Hal ini menghasilkan nilai rasio kompresi dengan perhitungan $(1500 \cdot 300 + 300 + 300 \cdot 1500) / (1500 \cdot 1500) = 0.4001$ atau 40.01%.



Kompresi terhadap instagram.png 450x450 dengan persentase $r = 10\%$

Eksperimen kedua adalah terhadap instagram.png ukuran 450x450 dengan persentase $r = 10$ (10% dari 423 adalah 45). Kali ini, gambar memiliki kanal alpha yang harus diperhitungkan. Setelah melalui svd, diperoleh tiga matriks berukuran 450x45, 1x45, dan 45x450. Nilai kompresi dapat dihitung, yakni $(450 \cdot 45 + 45 + 45 \cdot 450) / (450 \cdot 450) = 20.02\%$

BAB V

PENUTUP

6.1 Kesimpulan

- a. Sebuah gambar digital dapat direpresentasikan menjadi sebuah matriks $M \times N$. Untuk gambar yang berwarna, merupakan bentukan dari matriks warna merah, biru, dan hijau. Menggunakan Singular Value Decomposition, matriks dari gambar dapat diolah sedemikian sehingga menjadi 3 bagian matriks. Tiga bagian matriks tersebut terdiri dari matriks ortogonal U , matriks diagonal Σ , dan matriks ortogonal transpose V^T .
- b. Dari sub-matriks yang terbentuk dari matriks gambar, setelah menyimpan nilai singular yang diinginkan, tersimpan memori yang berpotensi lebih kecil dari memori awal.
- c. Submatriks yang terbentuk dapat direkonstruksi ulang menyerupai gambar awal. Semakin besar nilai r (semakin banyak nilai singular yang disimpan) maka gambar hasil rekonstruksi semakin menyerupai gambar awal. Oleh karena itu, metode SVD dapat diaplikasikan pada proses kompresi gambar.

6.2 Saran

- a. Program baru bisa menerima beberapa format file saja, dan format file tersebut digeneralisir menjadi mode RGBA untuk png dan RGB untuk selain png. Bisa ditingkatkan dengan menerima lebih banyak format file.
- b. Bisa dilengkapi dengan fitur renaming gambar setelah dikompresi

6.3 Refleksi

Dalam pengerjaan, terdapat beberapa kendala dalam memilih cara mendapatkan eigen value dan eigen vector, serta metode svd karena banyaknya cara yang memungkinkan. Namun demikian, tugas besar ini melatih penulis untuk mengimplementasikan pelajaran dan konsep yang telah didapat dari kuliah Aljabar Linear dan Geometri.

REFERENSI

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18- Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf>

http://www.math.utah.edu/~goller/F15_M2270/BradyMathews_SVDImage.pdf

<https://ristohinno.medium.com/qr-decomposition-903e8c61eaab>