Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri Aplikasi Nilai Eigen dan Vektor Eigen dalam Kompresi Gambar Semester I Tahun 2021/2022

KELOMPOK GWF

Anggota Kelompok :

Muhammad Gilang R.	13520137
Muhammad Fahmi Irfan	13520152
Willy Wilsen	13520160

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 JPG 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 1.1 Contoh kompresi gambar dengan berbagai tingkatan

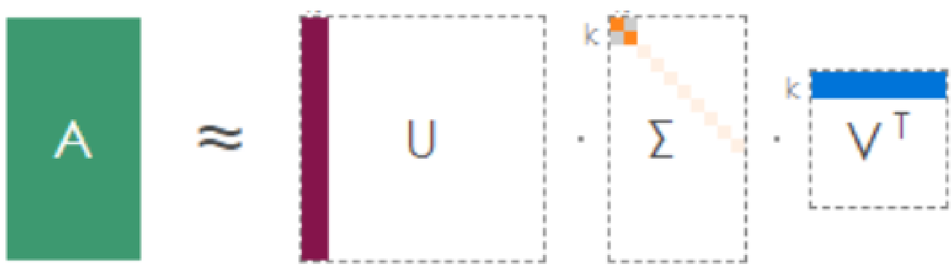
Sumber : *Understanding Compression in Digital Photography (lifewire.com)*

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 orthogonal U, matriks diagonal S, dan transpose dari matriks orthogonal V. Dekomposisi matriks ini dapat dinyatakan sesuai persamaan berikut.

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

Gambar 1.2 Algoritma SVD

Matriks U adalah matriks yang kolomnya terdiri dari vektor eigen otonormal 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.

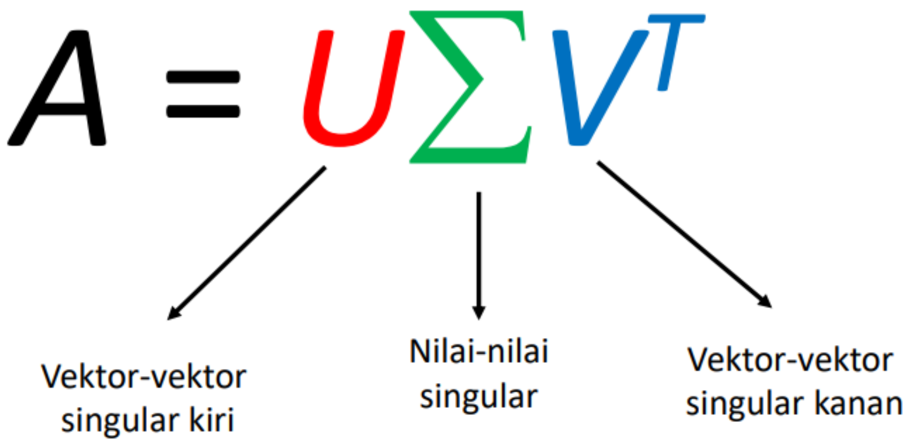


Gambar 1.3 Ilustrasi Algoritma SVD dengan rank k

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.

BAB 2 Teori Singkat

Singular Value Decomposition (SVD) adalah suatu metode yang bisa dimanfaatkan untuk mendekomposisikan suatu matriks A menjadi tiga komponen matriks U, Sigma, dan V^T.



Gambar 2.1 Formula dari *Singular Value Decomposition*

1. Vektor-vektor singular kiri (U)

   Bisa dicari dengan dari persamaan berikut.

   1. Cara 1

      $$(\lambda I - AA^T)x = 0 \ \rightarrow \det(\lambda I - AA^T) = 0$$

      Didapat vektor eigen $u_1, u_2, u_3, \ldots, u_k$ yang apabila dinormalisasi diperoleh $\widehat{u_1}, \widehat{u_2}, \widehat{u_3}, \ldots \ , \widehat{u_k}$

   2. Cara 2

      Untuk suatu i $\in \{1,2,3, \ldots, k\}$, vektor-vektor singular kiri dapat dicari dengan persamaan sebagai berikut.

      $$v_i = \frac{1}{\sigma_i} . A^T . u_i$$

2. Vektor-vektor singular kanan (V^T)

Bisa dicari dengan dari persamaan berikut.

1. Cara 1

$$(\lambda I - A^T A)x = 0 \ \rightarrow \det(\lambda I - A^T A) = 0$$

   Didapat vektor eigen $v_1, v_2, v_3, \dots, v_k$ yang apabila dinormalisasi diperoleh $\widehat{v_1}, \widehat{v_2}, \widehat{v_3}, \dots\ , \widehat{v_k}$

2. Cara 2

   Untuk suatu i $\in \{1,2,3, \dots, k\}$, vektor-vektor singular kiri dapat dicari dengan persamaan sebagai berikut.

$$u_i = \frac{1}{\sigma_i}\ .\ A\,.\,v_i$$

3. Nilai-nilai Singular (Σ)

   Untuk $\lambda_1, \lambda_2, \dots, \lambda_k$ nilai eigen yang tidak nol dari $A^T A$ yang berkoresponden dengan V atau AA^T yang berkoresponden dengan U, maka diperoleh :

$$\sigma_1 = \sqrt{\lambda_1}, \sigma_2 = \sqrt{\lambda_2}, \sigma_3 = \sqrt{\lambda_3}\ , \dots, \sigma_k = \sqrt{\lambda_k}$$

   Dengan $\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_k$ merupakan komponen-komponen nilai singular (Σ)

Dalam pencarian Komponen-komponen SVD tersebut, bisa digunakan algoritma **Power Method** atau **Power Iteration**. Algoritma tersebut dapat meng-*approxiomate* nilai dari Eigenvalues dan Eigenvektor pada suatu matriks, yaitu sebagai berikut.

Untuk suatu x_0 yaitu random vektor, dapat dilakukan iterasi sebanyak ***k*** kali, sehingga diperoleh:

$$x_1 = Ax_0$$

$$x_2 = AAx_0 = A^2 x_0$$

….

$$x_k = A^k x_0$$

Semakin besar ***k*** yang digunakan maka semakin baik nilai *approxiomate* yang dihasilkan.

Untuk suatu vektor *v*, pada pencarian eigenvaluenya bisa digunakan persamaan sebagai berikut.

$$\lambda = \frac{v^T A^T v}{v^T v}$$

Dalam konteks pemrogramman, persamaan tersebut dapat dipandang sebagai relasi rekurens dari

$$x_k = Ax_{k-1}$$

BAB 3 Implementasi Pustaka dan Program dalam Python

Operasi Primitif

1. Fungsi matriks diagonal
   def DiagonalMatriks(A):
   { Fungsi yang mengembalikan diagonal matriks A }
2. Fungsi transpose matriks
   def transpose(A):
   { Fungsi untuk mentransposekan matriks }
3. Fungsi Dot Matriks
   def Dot(A, B):
   { Fungsi untuk melakukan perkalian dot pada dua matriks }
4. Fungsi Outer dua matriks
   def Outer(A, B):
   { Fungsi untuk mengembalikan outer dua matriks }
5. Fungsi Identity matriks
   def Identity(A):
   { Fungsi untuk mengembalikan identity dari suatu matriks }
6. Fungsi Normalisasi matriks
   def Norm(A):
   { Fungsi untuk mengembalikan hasil normalisasi dari suatu vektor }
7. Fungsi untuk mengcopy matriks
   def CopyMatriks(A):
   { Fungsi untuk mengcopy dari suatu matriks ke matriks lain }
8. Fungsi size matriks
   def UkuranMatriks(A):
   { Fungsi untuk mencari ukuran dari matriks misalnya M, N }
9. Fungsi Zero Like
   def Zero_Like(A):
   { Mengembalikan sebuah array yang bernilai nol dengan size yang sama dengan matriks yang diinput }
10. Fungsi Array Of Float

def ArrayOfFloat(A):

{ Fungsi untuk mengassign array inputan sebagai array of float }

Operasi Untuk Kompressi

1. Fungsi Refleksi Householder

   def Refleksi_Householder(Matriks_Inputan) :

   { Mendekomposisi Matriks Menggunakan Householder Reflection yaitu menjadi Sigma dan Q}
2. Fungsi QR Eigen

   def qr_eigen(Matrix, maxIter) :

   { Menghitung nilai Sigma dan Q (bisa U ataupun V) }
3. Fungsi mencari matriks singular kiri (*U*)

   def Get_V(Matriks, Sigma, U) :

   { Menghitung matriks singular kiri }
4. Fungsi mencari matriks singular kanan (*V*)

   def Get_U(Matriks, Sigma, V) :

   { Menghitung matriks singular kanan }
5. Fungsi mencari Singular Value Decomposition (*SVD*)

   def SVD(A, K) :

   { Mendekomposisi Matriks A dengan iterasi sebanyak K }
6. Fungsi untuk menampilkan gambar dari suatu image

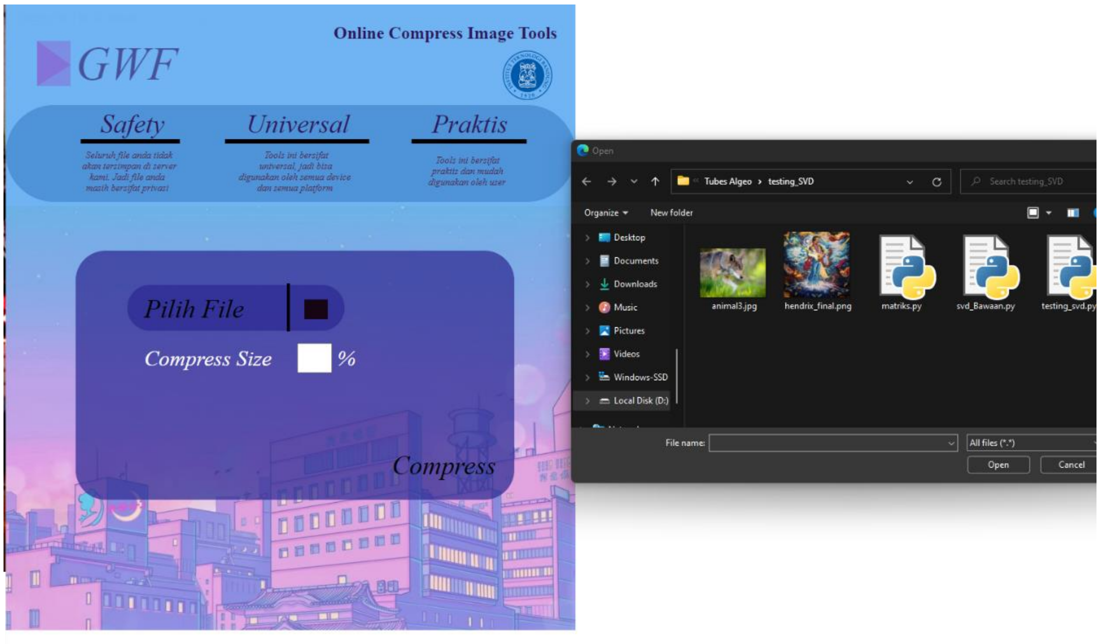
   def show_images(img_name):

   { Menampilkan image dari path img_name }
7. Fungsi untuk melakukan pemrosesan kompresi gambar

   def Kompress_image(Nama_image, k):

   { Melakukan kompresi gambar dari path Nama_image dengan iterasi K }

BAB 4 Eksperimen



Gambar 4.1. Halaman Utama Web Kompresi

Tabel 4.1.

Hasil Percobaan	
Mengkompresi gambar	
Before	After
	Dengan mengkompresi gambar sebesar 50% didapatkan gambar sebagai berikut.

	
	Dengan mengkompresi gambar sebesar 25% didapatkan gambar sebagai berikut. 
	Dengan mengkompresi gambar sebesar 10% didapatkan gambar sebagai berikut. 

BAB 5 Kesimpulan

A. Kesimpulan

Standar Value Decomposition (SVD) merupakan satu teknik yang bisa digunakan untuk mendekomposisi (memfaktorkan) sebuah matriks. Pada aplikasinya, SVD dapat digunakan untuk mengkompress suatu gambar. Selain SVD, sebenarnya terdapat banyak algoritma lain yang bisa dipakai. Namun, diantara algoritma lain, SVD merupakan yang paling mudah diterapkan. Dalam penerapannya, pembuatan algoritma SVD bisa memanfaatkan *Power Iteration* atau *Power Method*, yaitu dengan mengiterasi nilai eigen value dan vektor eigen dari suatu matriks. Semakin banyak iterasi yang dihasilkan maka semakin baik kualitas gambar yang diperoleh.

B. Saran Pengembangan

Python merupakan salah satu Bahasa pemrogramman yang lebih mudah dipakai oleh banyak orang karena library yang disediakan cukup lengkap. Namun, dalam kinerjanya, terutama dalam pengolahan kompressi gambar, python lebih lambat dalam pemrosesannya jika dibandingkan bahasa pemrogramman lain.

C. Refleksi

Dengan adanya tugas besar 2 Aljabar Linear dan Geometri, dapat melatih skill programing yang dimiliki masing-masing individu serta kerja sama atau dan kolaborasi antar anggota dalam kelompok. Dengan adanya keterbatasan waktu, Kami telah berusaha secara maksimal untuk menyelesaikan tugas yang diberikan.

REFERENSI

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

Slide Bahan Ajar Mata Kuliah IF2123 Aljabar Linier dan Geometri oleh Rinaldi Munir

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2021-2022/algeo2122.htm