SOFTWARE PROJECT REPORT

AI25BTECH11017-BALU

0.1 AIM:

Write a program that compresses image using truncated SVD(low-rank approximation)

0.2 OBJECTIVE:

constructing a program that compresses image.we turned an image into 2D-array(matrix) of pixels and compressed it using truncated SVD by JACOBI'S algorithm implementing in c-language.

0.3 PROCEDURE:

1) Taking image as 2D-array of pixels
2) Apply truncated SVD on that matrix using JACOBI'S algorithm

0.3.1 EXPLANATION OF JACOBI'S ALGORITHM:

3) a) let us say the matrix to be A
   b) $\mathbf{A} = \mathbf{u}\sigma\mathbf{v}^T$
   c) calculate $\mathbf{A}^T\mathbf{A}$
   d) $\mathbf{A}^T\mathbf{A} = \mathbf{v}\sigma^2\mathbf{v}^T$
   e) we find largest off-diagonal element and wants to make it zero
   f) we will cook a matrix $\mathbf{R}(p, q, \theta) = \mathbf{I}$ except $R_{pp} = R_{qq} = \cos\theta$ and $R_{pq} = \sin\theta$ and $R_{qp} = -\sin\theta$
   g) we will update $\mathbf{A}$ as $\mathbf{R}^T\mathbf{A}\mathbf{R}$
   h) repeat this untill all off-diagonal elements become zero
   i) the diagonal elements are eigen values of $\mathbf{A}^T\mathbf{A}$
   j) $\sigma_i = \sqrt{\lambda_i}$
   k) $\mathbf{v}$ is product of all $\mathbf{R}$'s and $\mathbf{u}_i = \frac{\mathbf{A}\mathbf{v}_i}{\sigma_i}$
   l) then we will take top k singular values and output $\mathbf{A}_k = \mathbf{u}_k\sigma_k\mathbf{V}_k^T$
4) we will turn $\mathbf{A}_k$ into a PGM file and then convert into .png or .jpg

0.4 algorithm explanation and pseudo code:

we will be having a matrix let us say it to be $\mathbf{A}$,it singular value decomposition can be written as

$$\mathbf{A} = \mathbf{u}\sigma\mathbf{v}^T \tag{4.1}$$

$$\mathbf{u}^T\mathbf{u} = \mathbf{I} \ \mathbf{v}^T\mathbf{v} = \mathbf{I} \ \sigma^T = \sigma \tag{4.2}$$

where $\mathbf{u}$ and $\mathbf{v}$ are orthogonal matrices and σ is diagonal matrix

$$\mathbf{A}^T\mathbf{A} = \mathbf{v}\sigma\mathbf{u}^T\mathbf{u}\sigma\mathbf{v}^T = \mathbf{v}\sigma^2\mathbf{v}^T \tag{4.3}$$

it is in the form of eigen value decomposition and we can say $\sigma_i = \sqrt{\lambda_i}$ now we will construt a matrix $\mathbf{R}(p, q, \theta)$ to find eigen values of $\mathbf{A}^T\mathbf{A}$
now we will first find largest off-diagonal element(a_{pq}) and by updating $\mathbf{A}$ in this fashion $\mathbf{R}^T\mathbf{A}\mathbf{R}$ we will make $a_{pq} = 0$ and repeat the process till all off-diagonal elements become zero
constructing $\mathbf{R}$
$\mathbf{R}(p, q, \theta) = \mathbf{I}$ except $R_{pp} = R_{qq} = \cos\theta$ and $R_{pq} = -\sin\theta$ and $R_{qp} = \sin\theta$ where

$$\tan 2\theta = \frac{2a_{pq}}{a_{qq} - app} \tag{4.4}$$

now we will get a nearly diagonal matrix that is σ^2 and $\mathbf{v}$ is product of all $\mathbf{R}$ and we can get $\mathbf{u}$ as $\mathbf{u}_i = \frac{\mathbf{A}\mathbf{v}_i}{\sigma_i}$ where $\mathbf{u}_i$ and $\mathbf{v}_i$ is singular vectors
now we take top-k singular values and k singular vectors and construct

$$\mathbf{A}_k = \mathbf{u}_k\sigma_k\mathbf{v}_k^T \tag{4.5}$$

0.5 GILBART-STRANG SUMMARY

The video explains about Singular Value Decomposition(SVD), a final and best way to factorise a matrix.
$\mathbf{A} = \mathbf{u}\sigma\mathbf{v}^T$
where $\mathbf{u}$ and $\mathbf{v}$ are orthogonal matrices,σ is a diagonal matrix.
Now to find SVD the trutor use the fact that $\mathbf{A}^T\mathbf{A} = \mathbf{v}\sigma^2\mathbf{v}^T$
Here $\mathbf{A}^T\mathbf{A}$ is symmetric and positive semi-definte matrix.
$\mathbf{A}^T\mathbf{A} = \mathbf{v}\sigma^2\mathbf{v}^T$ this is just eigen value decompostion of $\mathbf{A}^T\mathbf{A}$ he uses this fact to singular values and singular vectors of $\mathbf{A}$.then we can $\mathbf{u}$ from $\mathbf{A} = \mathbf{u}\sigma\mathbf{v}^T$ this equation.

0.6 RECONSTRUCTED IMAGES

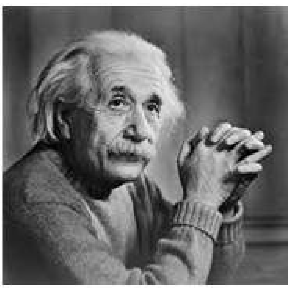

Fig. 4.1: original einstein image

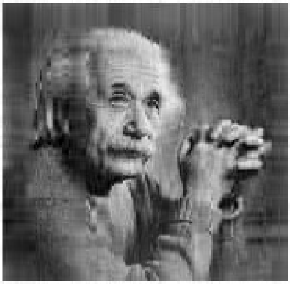

Fig. 4.2: reconstructed image k=45

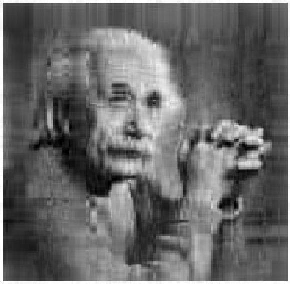

Fig. 4.3: einstein k=25

0.7 ERROR ANALYSIS

0.8 Results for Different k

k	Compression Ratio
45	0.1002
25	0.1129

TABLE 4: Einstein

k	Compression Ratio
15	0.2111
100	0.1904
500	0.0979

TABLE 4: Globe

k	Compression Ratio
100	0.1370
400	0.0818
850	0.0061

TABLE 4: Greyscale

0.9 pseudo code

Input: PGM image file name

Output: Reconstructed image with reduced rank

Read PGM image using **read_pgm**:

- Open file and read magic number (P2 or P5)
- Skip comments starting with '#'
- Read image width, height, and max value
- Read pixel values into matrix $A[m \times n]$

Compute SVD using **compute_svd**(A):

1) Compute $A^T A$
2) Apply Jacobi's algorithm to find eigenvalues and eigenvectors of $A^T A$
3) Compute singular values $\sigma_i = \sqrt{\lambda_i}$
4) Sort singular values in descending order and reorder columns of V
5) Compute $U = AVS^{-1}$

Ask user for truncation rank k

Reconstruct compressed image using:

$$A_k = \sum_{i=1}^{k} \sigma_i U_i V_i^T \tag{5.1}$$

Write reconstructed image using **write_pgm**:

- Write header (P2, width, height, maxval)
- Write pixel values from A_k

Compute errors:

$$E = ||A - A_k||_F \tag{5.2}$$

$$\text{Relative error} = \frac{E}{||A||_F} \tag{5.3}$$

Display:

- Reconstructed image filename
- Frobenius and relative errors

Free all allocated memory

0.10 performance of algorithm

k	**size**
45	19.3kb
25	18.5kb

TABLE 5: Einstein

The size of original image was 20kb. we can see that for k=25 the size 18.5kb so we reduced 1.5kb with minimum quality of image.also the relative error also good,so our algorithm works good