Grayscale Image Compression With Principal Component Analysis (PCA)

Marcos Daniel Calderón-Calderón

March 7, 2019

1 Grayscale Image Compression using Principal Component Analysis (PCA)

Let $I_{m\times n}$ a grayscale image with m rows and n columns. It is necessary to choose a square filter of size k, $(F_{k\times k})$. The purpose of the filter is to divide the original image into fragments of size $k \times k$. In conclusion, the image will now be represented as a signal of $\frac{m}{k} \times \frac{n}{k}$ fragments.

Now, the original image is rearranged in a signal:

$$X_{\left(\frac{m}{k}\times\frac{n}{k}\right)\times(k\times k)} \tag{1}$$

The fragments of the original image are chosen from left to right, from top to bottom. Every fragment is rearranged in one of the $\left(\frac{m}{k} \times \frac{n}{k}\right)$ rows of matrix X, in the same way, pixels of specific fragment are chosen from left to right, from top to bottom. The rows of matrix X are chosen from top to bottom.

Pixels of matrix X do not represent centered data, therefore, a new vector:

$$M_{(k\times k)\times 1} \tag{2}$$

is filled with the mean of every column of matrix X. The next step is to subtract the mean in each element of the corresponding column:

$$X_{(i,j)} = X_{(i,j)} - M_{(j,1)} \tag{3}$$

thus, the data of the matrix X will be centered.

A new matrix C is calculated with the following operation:

$$C_{\left(\frac{m}{k}\times\frac{n}{k}\right)\times\left(\frac{m}{k}\times\frac{n}{k}\right)} = XX^T \tag{4}$$

in this case, C is te covariance matrix.

Eigenvectors and eigenvalues of matrix C are calculated with an specific method. In this case, the Jacobi's method was used. The main disadvantage of this method: in some cases it is slow.

$V_{\left(\frac{m}{k}\times\frac{n}{k}\right)\times 5}$ will be the matrix with the 5 most informative eigenvectors.

Finally, the compressed image will be built with the following matrix multiplication:

$$Z_{(k\times k)\times 5} = \left(X_{\left(\frac{m}{k}\times\frac{n}{k}\right)\times(k\times k)}\right)^T \left(V_{\left(\frac{m}{k}\times\frac{n}{k}\right)\times 5}\right) \tag{5}$$

To obtain the uncompressed image, the operation is very easy:

$$X_{\left(\frac{m}{k}\times\frac{n}{k}\right)\times(k\times k)} = \left(\left(Z_{(k\times k)\times 5}\right)\left(V_{\left(\frac{m}{k}\times\frac{n}{k}\right)\times 5}\right)^T\right)^T \tag{6}$$

To obtain the uncompressed image, the data of the matrix X are rearranged to the structure of the original image $I_{m\times n}$. Do not forget that the vector $M_{(k\times k)\times 1}$ must now be added to the pixels of the corresponding columns for matrix X.

2 Results

Figure 1: Original image.

Figure 2: Uncompressed image with $k = 5$ (25 pixels per fragment).

Figure 3: Uncompressed image with $k = 8$ (64 pixels per fragment).

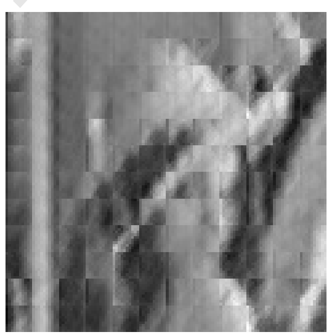

Figure 4: Uncompressed image with $k = 10$ (100 pixels per fragment).