الجامعة المصرية اليابانية للعلوم و التكنولوجيا
エジプト日本科学技術大学
EGYPT-JAPAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Three images depicting the three layers Y,C_r, and C_b components individually and the overlapping to produce the original picture.

Digital Images

Representation, Compression, and JPEG Format

In the background the Discrete Cosine Transform basis functions.

Math111-F20 Course report

Table of contents:

I. Introduction:

After the invention of computers, digital image usage and processing took over analog images. Images are important documents today. As we try to reach a more efficient method of storing and transmitting information in general, image processing and storing have rapidly developed over the last decade. One of the earliest applications of digital image, in the early 1920s, can be seen in the newspaper industry. It was about the pictures that were sent by submarine cable between London and New York, known as Bartlane cable picture transmission system. At that time, it took around three hours to send one image with 5 gray levels. Today, we sent and receive hundreds of digital images through our electronic devices, and we view tens of images on websites daily. Of course, such advancement would not have been impossible without the advancement and modern methods used in image processing and compression.

II. Different ways to represent an image as a matrix:

If you could bring an image and keep zooming in, you will reach a limit, where you will be able to see only tiny blocks, which are called pixels. The word pixel is a combination of pix (from "pictures", shortened to "pics") and el (for "element"), and it is the limit as it's the smallest unit of building images, so pixels are considered to be the most basic building blocks for the digital images; controlling the pixels leads to controlling in the image itself.

You can imagine any image as a matrix A of some elements C, so you have a specific number of columns and rows for it; let it be a N*M matrix, where the N is the number of rows, and M is the number of columns; each element of the matrix A represents a pixel, and the number that's given for each pixel (the element of the matrix) is the value of the pixel, defining its intensity; as shown in Figure 1; the image consists of only two colors, so it'll be a binary type image as there are only two values (0,1) needed to offer the colors of the image, where the value of the white is (1). The black is (0) colors, or (1) can be considered to be the value of true and (0) to be the value of false. This type or representation needs only one bit for each pixel.

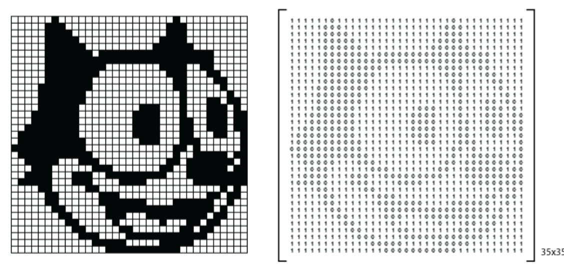

Figure 1 A binary type image

Consider another case, where the image has no longer two colors only, and instead, it has a certain intensity of a color in each pixel as shown in Figure 2. so how is the approach that will be used this time. The type of this image is named a monochromatic image; each pixel is going to be given a value, ranging from 0 to 255, where 255 represents the utmost intensity (the white color), and 0 represents the minimum intensity (the black color) giving a total of $256 = 2^8$ different levels of gray which implies that each pixel is an 8-bit size or 1-byte. If you need to represent a pixel with a 25% of white intensity, you need to divide the total number of values (256) by four as 0 also represents a value for the black color, so a 25% white intensity has a value of 256/4, which gives 64; what if you want to calculate a more irrational percentage of white intensity, so what are the

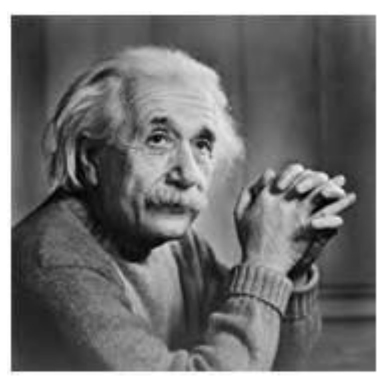

Figure 2 A grayscale portrait of Albert Einstein

steps that are going to be followed?; it's known that 100% is the maximum intensity (White color), so 17% of black means 87% of white, so you'll use the same approach again by multiplying (87/100) * 256 and then rounding up to give the value 223, so 13%-black pixel has a value of 223, and so does an 87%-white pixel.

Suppose as this time that you have a picture with many colors; that type of images is said to be represented by the RGB format, which means that each pixel has three different mixed intensities of red, green and blue colors to produce different colors. Color images can be represented by three matrices, each matrix specifies the amount of Red, Green and Blue that makes up the image. This color system is known as RGB. The elements of these matrices are integer numbers between 0 and 255, and they determine the intensity of the pixel with respect to the color of the matrix. Thus, in the RGB system, it is possible to represent $256^3 = 2^{24} = 16777216$ different colors. Also, this implies that a single RGB pixel is 24-bit size or 3-byte size with every byte to represent a dimension or color component.

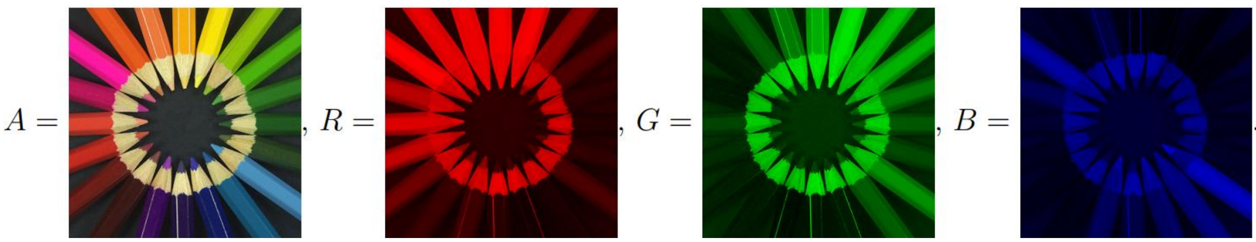

Figure 3 A colored image(A) is shown and its three Red(R), Green(G), and Blue(B) components

Notice that the colorful image in Figure 3 is simply a mix of three other images, with the RGB colors, but with different intensities, helping the colorful image to give it its appearance; this time, the white color will be represented by three maximum values of 255 for blue, 255 for green and 255 for red, so by intuition, the white represents maximum intensity, and the black represents the minimum intensity, which can be represented by three minimum values of 0 for blue, 0 for green and 0 for red.

There is a point that might be concluded intuitively as whenever you find a pixel having the same intensity values for the 3 RGB colors; it will give a certain intensity of the gray color as there is no color, having a value higher than the other two colors to dominate them.

A more advanced application of representing pixels by matrices is that the image transitions as seen in PowerPoint© presentations; assuming, you've got two images of an equivalent size of Rows and columns, represented by two different matrices A and B (only difference in the value of each pixel as they are two different-colors images; in case of RGB format, different three values of pixels in each matrix); you then have a matrix $M(t) = (1-t)A + t B$, thus $M(0) = A$, and $M(1) = B$, when the value of t lies in the interval [0, 1], which means that the value of M(t) varies from matrix A to be matrix B with t varying from 0 to 1, showing the transition between slides effect as in PPT presentations; consider at this point that you have a colorful image; therefore the same approach is going to be used, but at this point, it'll be used three times as for every dimension (matrix) of the RGB colors.

III. What's image compression:

Image compression is the encoding of a digital image so that it takes fewer bits while preserving the quality of the image. Its objective is to reduce the size so it can be stored or transmitted in a more efficient form. Image compression can be mainly divided to two main categories: **Lossless Compression**, and **Lossy Compression.**

First: Lossless Compression

In Lossless Image Compression a class of data compression algorithms are applied to the original image, which is then encoded in a more efficient method to reduce its size, however when decoded it has the exact same information as the original image. Lossless Compression results in a reconstructed data representation that exactly matched the original image.

How Lossless Compression works:

In this example we will use tables to represent images with every cell representing a pixel with 3-byte size where every byte specifying the intensity value of either red, green, or blue (RGB scale).

Red|Green|Blue
0:255|0:255|0:255

Figure 5: Example Cell

Image data are typically stored as a list of pixel values. For simplicity, we will assume that the 4x4 pixel image in Figure 6 will be stored as following:

(255,255,255), (255,255,0), (0,0,0), (255,255,0),

(255,255,0), (255,255,0), (255,255,0), (255,255,0),

(255,255,0), (255,255,0), (255,255,255), (255,255,255),

(255,255,255), (255,255,0), (255,255,0), (255,255,0).

255|255|255	255|255|0	0|0|0	255|255|0
255|255|0	255|255|0	255|255|0	255|255|0
255|255|0	255|255|0	255|255|255	255|255|255
255|255|255	255|255|0	255|255|0	255|255|0

Figure 6: A simple 4x4 image (packman)

Since this image contains 16 pixels and each pixel is 3 bytes of color data, this means that the size of this image is equal to 48 bytes. However, it is possible to store the same image with the exact color-pixel values in a smaller number of bytes (compression).

One method of this type of compression is Run-Length Encoding (RLE), in which the encoder reduces repeated or redundant information. In this method, we add an extra place-value before the color-value to represent the repetition of that color-value. Applying this method will produce the following:

(1,255,255,255), (1,255,255,0), (1,0,0,0), (7,255,255,0), (3,255,255,255), (3,255,255,0).

As you can notice, the number in black represents how many times the color code following it will repeat before moving on to the next color code. As this method suggest, we now need an extra byte to represent repetition. The size of the image when coded under this method is now 24 bytes instead of 48 bytes with no change in the color values and by producing the exact same image.

However, this method can result in a larger image size for rare cases where colors do not repeat which means we have added an extra byte in each pixel without reducing the number of pixels represented.

As shown in Figure 7, every pixel in this image is a different color, which means there are no repeated or redundant information to be reduced. Another example would be an alternating white and black pixels image.

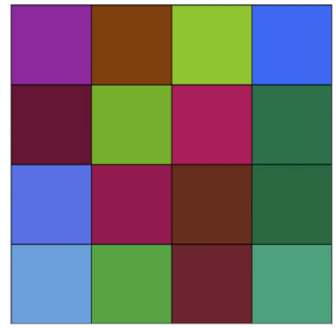

Figure 7: A 4x4 pixel image representing limitation to Run-Length Encoding

Methods of Lossless Compression:

- Run-length encoding – used in default method in PCX and as one of possible in BMP, TGA, TIFF
- Area image compression
- Predictive coding – used in DPCM.
- Entropy encoding – the two most common entropy encoding techniques are arithmetic coding and Huffman coding.
- Adaptive dictionary algorithms such as LZW – used in GIF and TIFF.
- DEFLATE – used in PNG, MNG, and TIFF
- Chain codes

Second: Lossy Compression

In Lossy Image Compression (irreversible compression) a class of data encoding methods that uses inexact approximations are applied to the original image to reduce its size, however when decoded it has different information than the original image, yet close to it and maybe with no visual difference to human eye. Lossy Compression results in a reconstructed data representation that does not match the original image and it is not possible to retrieve the original image after from a lossy compression.

How Lossy Compression works:

Another way of representing an image is YC_bC_r instead of RGB scale. The Y, C_b, and C_r components of one-color image are defined in YUV color coordinate, where Y is commonly called the luminance and Cb, Cr are commonly called the chrominance. The meaning of luminance and chrominance is described as follows:

Luminance: received brightness of the light, which is proportional to the total energy in the visible band.
Chrominance: describe the perceived color tone of a light, which depends on the wavelength composition of light chrominance is in turn characterized by two attributes – hue and saturation.

1. **hue:** Specify the color tone, which depends on the peak wavelength of the light.
2. **saturation:** Describe how pure the color is, which depends on the spread or bandwidth of the light spectrum.

The Y, C_b, C_r values in the YUV coordinate are related to the RGB values in the RGB coordinates by:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.334 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

Equation 1 Relation between YCbCr and RGB adapted from Wei, Wei-Yi. "An Introduction to Image Compression." (2008).

It is also possible to transform the YUV coordinate back to RGB coordinate as the inverse matrix can be calculated from Equation 1. This method of compression depends on the fact that human eyes are more sensitive to the luminance than to the chrominance. As shown in the following table you can notice the visual quality of the image with respect to the reduction in the Chrominance and Luminance.
Recommended to check the link in the caption under the table for a more interactive experience.

Reduction in Luminance			
Percent of reduction	50%	75%	90%
Reduction in Chrominance			
Percent of reduction	50%	75%	90%

Table 1 : Image with variant Luminance and Chrominance levels generated from this link: https://jakearchibald.github.io/image-experiments/channels/?demo=lane&uv=1. Copyrights to Jake Archibald.

Methods of Lossy Compression:

- Chroma subsampling. This takes advantage of the fact that the human eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information in the image.
- Transform coding – This is the most used method.
- Reducing the color space to the most common colors in the image. The selected colors are specified in the color palette in the header of the compressed image. Each pixel just references the index of a color in the color palette, this method can be combined with dithering to avoid posterization.
- Fractal compression method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Fractal algorithms convert these parts into mathematical data called "fractal codes" which are used to recreate the encoded image.

IV. Basic idea behind JPEG format:

Compression techniques have developed to satisfy the needs of the new applications requiring high quality imaging, these developments in the compression techniques took time and the arising compression methods were not compatible with each other, as the applications was being developed day by day and the number of applications increased dramatically there was a crucial need for a versatile image compression format that can be used to compress for all of these applications, moreover the nature of our human eye necessitates takin some certain aspects of images into consideration to reach higher levels of image compression, also these aspects let images be compressed in methods that adapt to its transmission, one of these compression formats is JPEG which is covered briefly in this section, this format is very popular as around 80% of images transmitted through the internet are using the JPEG standard, JPEG is an abbreviation for the committee that developed it "The Joint Photographic Experts Group", it was made by members of the International Organization for Standardization (ISO) and the International Telecommunications Union (ITU), at that time the computers of normal consumers could display and manipulate full color photographs as they had sufficient processing power, nevertheless, these photographs needed a huge amount of bandwidth when sent over a network connection, also saving a local copy of the photograph took just as much space, the problem with other compression formats was the major tradeoffs they made like having one of the two issues either tremendous amount of data loss in the image or low amounts of compression, however, JPEG could compress photographs with high compression ratios the least amount of data loss, and regardless of the popularity of format, JPEG committee members addressed some problems in the format and worked on developing these issues creating various implementations for JPEG compression as SVD, Haar wavelet, quantization and Huffman coding, this section addresses some of the basics of JPEG algorithm mentioning quantization and Huffman coding.

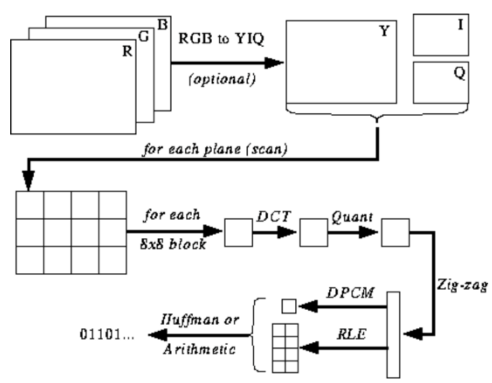

Figure 8 JPEG Encoding Cycle

The JPEG format is excellent in compression of full color photographs (24 bit) and also grayscale photographs that have various shades of gray, this is due to its algorithm and nature of compression, however, you will not feel the same if you used JPEG with web graphics, scanned text or other images where edges of the objects have sharp transitions, the format also features an alterable compression ratio making the user determine the size and quality of his final image, by that photographs can be highly compressed with less quality or leave off high compression resulting in a photo that cannot be distinguished from the original, compression and decompression of JPEG consists of 4 independent phases, first, dividing the image into 8 by 8 pixel blocks, second, converting the information from the spatial domain into the frequency domain by applying discrete cosine transform to every block, third, quantizing the frequency information to get rid of unnecessary information, fourth, compressing the final bit stream using standard compression techniques.

Phase 1:

Compressing an entire image is not efficient way and would not yield to a great result, so it would be a nice way to prepare an image to be compressed by color space transformation and dividing.

Color space transformation: in gray scale, every 8 bits in the image represents a pixel with a value (0-255) per pixel, otherwise, colored images in RGB scale have 3 bytes per pixel each byte represent the red, green or blue component with values (0-255). To make the operations less complex, JPEG will convert from RGB color space to YCbCr color space, in which Y component represents luminosity of the pixel, Cb and Cr are the chrominance values for red and blue.

Dividing: JPEG divides the image into blocks, each block is represented with 8x8 matrix. Suppose an image that is 1024 x 720 pixel, dividing this image will produce a 128 x 90 blocks, each block is 8 x 8 pixels. If the image's pixels are not a multiple of 8, extra pixels are added to the bottom or right or both to reach the next multiple of 8.

Phase 2:

Human vision has a drop-off at higher frequencies, removing them from image will affect the image space and it would seem the same for human eye. Transforming the information from the spatial domain to frequency domain in order to know which part is less important than others.

Firstly, the scale of the pixels has to be centered on zero, so the value 128 will be subtracted from each byte. The scale will change from (0-255) to (-128-127).

Many algorithms are used to transform from the spatial to frequency domain, the most used one is the Fast Fourier Transform. A JPEG is a special case that the information does not contain imaginary components, so a faster method is used "Discrete Cosine Transform" DCT.

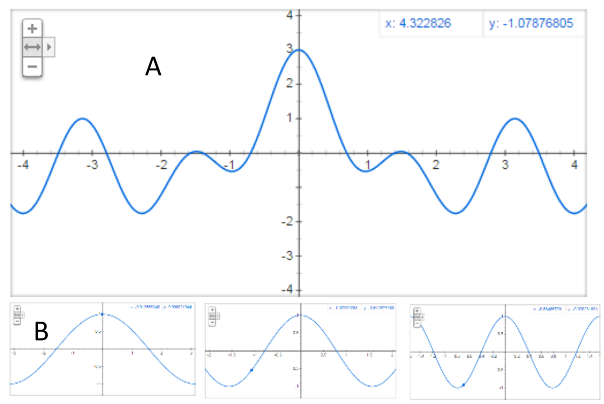

Figure 9: signal A is formed from functions B.

The idea of DCT is forming any signal of image from a combination of cosine functions, for example in one dimension (figure 9) the signal A is formed form the summation of the functions B.

Any 8 x 8 block can be represented as a sum of weighted cosine functions with numerous frequencies, the basic functions are shown in figure 10:

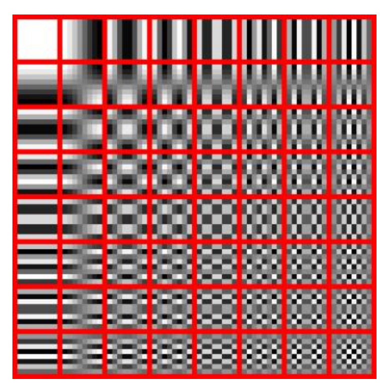
Figure 10: basis cosine functions, vertical frequency increase from left to right, horizontal frequency increase from top to bottom

using the following formula and the basis function will generate a new 8 x 8 matrix that has a scale between (-1024-1023) represent the weights that will be used to reconstruct the image again.

$$G_{u,v} = \frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y}\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

Equation 2

The upper left of the matric is called the DC coefficient, and it is the larger magnitude. The remaining 63 values is called the AC coefficient.

Phase 3:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Figure 11: Standard Quantization matrix

Dividing the frequency domain matrix by a constant (Quantization) matrix that define how much an image would be reduced.

Each component in the frequency matrix, resulted from phase two, is divided by a corresponding component in quantization matrix, as the corresponding value is getting bigger the result will approach zero and that reduce the amount of information for this pixel. The quantization matrix depends on the amount of compression wanted, for example, the standard quantization matrix in figure 11 for 50% compression. The result is rounded to the nearest integer.

Phase 4:

After quantization, an 8 x 8 matrix is resulted with most of its values zeros and the JPEG collect all zeros together to compress the data in the block.

JPEG transform the 8 x 8 matrix into a 1 x 64 vector, starting from top left value and ordered zigzag diagonally (figure12) to the bottom right value.

Huffman encoding is used to define each value except for the DC value, every value is represented as (run-length, size) (value), where run-length is the number of zeros preceding the value and size is the number of bits needed to represent the value. The idea is the more frequent value have a less bits than infrequent values.

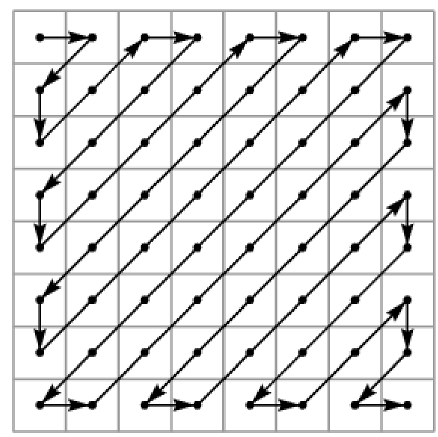

Figure 12: zig zag order

DC values is encoded by comparing each block's DC value with the preceding one in zig-zag manner also and storing the bits required for the difference value. For example, 2 adjacent DC values have a difference -4 the JPEG store the size 3 bits (the actual value stores is the binary value 100) that gives the image a higher compression.

V. References:

Archibald, J. (2020, July 16). Image compression deep-dive. Retrieved January 26, 2021, from https://youtu.be/F1kYBnY6mwg

Image Compression. (n.d.). Retrieved January 28, 2021, from https://communications.uwo.ca/comms/web_design/image-guidelines/compression.html

Laxman, S., S., & Shankar, P., A. (2020, August). Cryptography and Image Processing by Matrices. Retrieved January 27, 2021, from https://www.irjet.net/archives/V7/i8/IRJET-V7I8746.pdf

Matrices and Image Transitions. (n.d.). Retrieved January 27, 2021, from http://www.cdme.im-uff.mat.br/matrix/matrix-html/matrix_morph/matrix_morph_en.html

N, S., & Unnithan, S. (2017). *Image Compression Techniques: A Review, 5*(1).

Nectoux, A. (2015, May 27). Matrices and Digital Images. Retrieved January 25, 2021, from http://blog.kleinproject.org/?p=588

Razafindradina, H. B., Randriamitantsoa, P. A., & Razafindrakoto, N. R. (2016). Image Compression with SVD : A New Quality Metric Based On Energy Ratio. Retrieved January 26, 2021, from https://arxiv.org/ftp/arxiv/papers/1701/1701.06183.pdf

Types of Images. (n.d.). Retrieved January 24, 2021, from https://www.tutorialspoint.com/dip/types_of_images.htm

Verma, S., & Krishna, J. P. (2013, November 15). Image Compression and Linear Algebra. Retrieved January 26, 2021, from https://www.cmi.ac.in/~ksutar/NLA2013/imagecompression.pdf

Wei, W. (2008). *An Introduction to Image Compression*.