<u>Understanding Matrices to Perform Basic Image Processing on Digital Images</u>

Traditional photography has been fading away for decades with the introduction of digital image sensors. The majority of photography today is done using digital cameras due to the convenience of being able to instantly access the images without the hassle of having film processed. Digital images can be created in a variety of ways outside of using digital cameras as well; scanning printed images and using programs such as MS Paint (a basic example) can be other ways to create a digital image.

Digital images can be manipulated using matrices, since when broken into the smallest bits of information (pixels) they are matrices. Figure 1 demonstrates binary pixel art, in which each box is being represented in the correlating matrix with either a "1" or a "0". A zero means the box contains nothing (would normally appear transparent or white), while a one indicates that the box is filled (in this case it is filled with black).
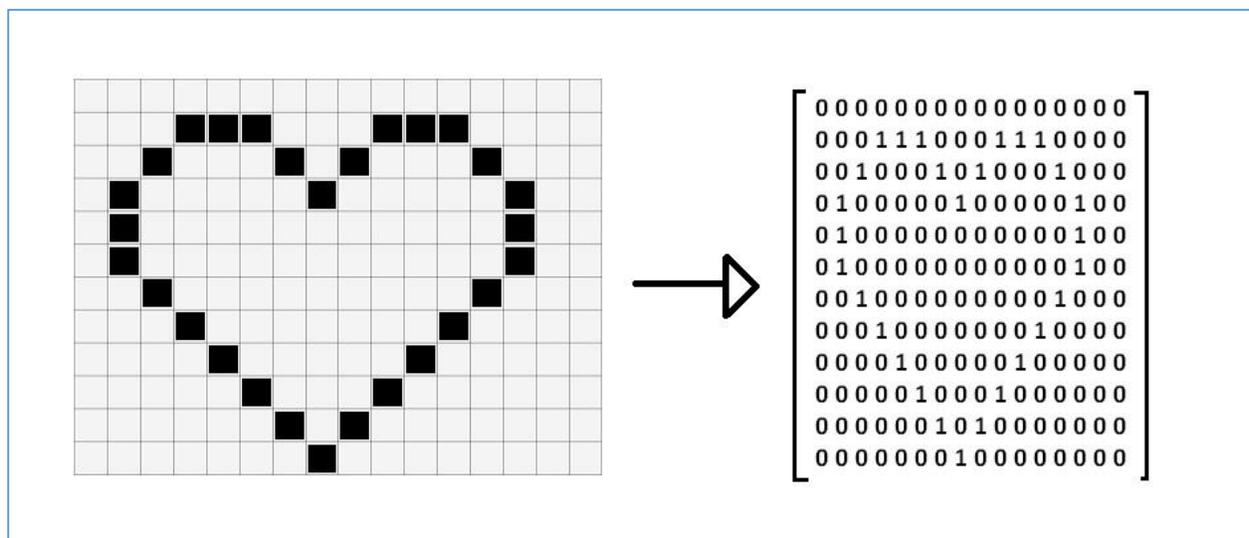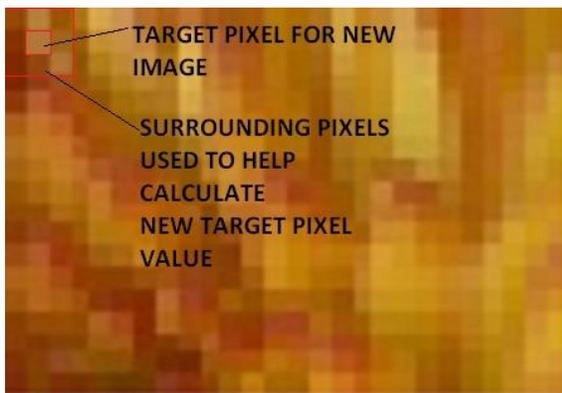

Figure 1: Binary pixel art

In normal digital images, the number that each spot of the matrix contains is much more complex and has information for the amount of red, green, and blue that the pixel should display. These numbers are represented using a system of hexadecimal numbers. In hexadecimal, black is #000000 (equivalent hexadecimal number for 0) and white is #FFFFFF (equivalent to red, blue, and green each containing the highest possible amount). So each pixel stores information about what color it should be displaying in the digital image, putting the output within a small 1x1 square. When viewed far away and together, they create the illusion of lines, curves, and blended colors instead of the individual squares that they are (see *Figure 2*).

Figure 2: a portion of a digital image zoomed in. You can see each square and that each one represents an individual color.

       The ability to break the image into data points has allowed for the creation of photo editing techniques that would be much more difficult (if not impossible) to apply in film photography. One popular method of doing this is utilizing matrix "kernels." The "kernel" for an affect is a small matrix, often size 3x3, that is then applied to each individual pixel of the digital image. Rather than using normal matrix multiplication, the kernel is applied by convolution. Each new target pixel color value is calculated using the original target pixel and the surrounding pixels. Below are two examples of kernel convolution.
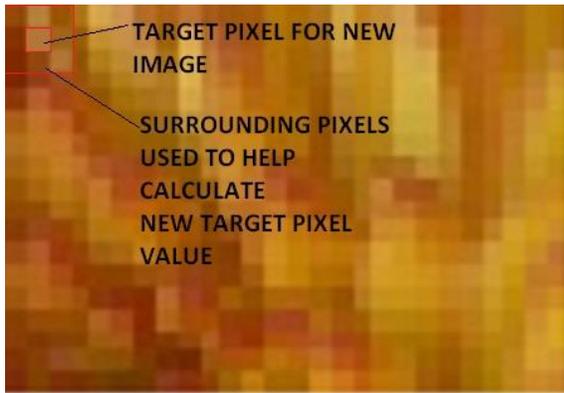

TARGET PIXEL FOR NEW IMAGE

SURROUNDING PIXELS USED TO HELP CALCULATE NEW TARGET PIXEL VALUE

PIXELS:

$$\begin{bmatrix} x1 & x2 & x3 \\ x4 & x5 & x6 \\ x7 & x8 & x9 \end{bmatrix}$$

BOX BLUR

KERNEL:

$$1/9 * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

NEW TARGET PIXEL VALUE =

$$(1/9)*(1*x1 + 1*x2 + 1*x3 + 1*x4 + 1*x5 + 1*x6 + 1*x7 + 1*x8 + 1*x9)$$

GAUSSIAN BLUR

PIXELS:

$$\begin{bmatrix} x1 & x2 & x3 \\ x4 & x5 & x6 \\ x7 & x8 & x9 \end{bmatrix}$$

KERNEL:

$$1/16* \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

NEW TARGET PIXEL VALUE =
(1/16)*(1*x1 + 2*x2 + 1*x3 +
2*x4 + 4*x5 + 2*x6 +
1*x7 + 2*x8 + 1*x9)

Many popular image processing effects are a result of kernel convolution. Here are some more examples of possible kernels:

ORIGINAL

SHARPEN

KERNEL:

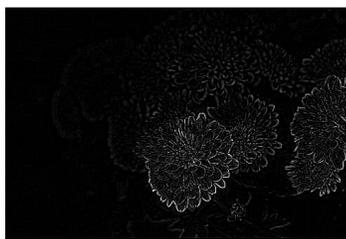$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

EMBOSS

KERNEL:

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

OUTLINE

KERNEL:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

The matrix representation of an image also allows for functions such as adding brightness or rotation. For brightness, a constant (such as 20) is added to the red, green, and blue values of each pixel. For rotation, a new image can be generated by moving the rows and columns of pixels around (see *Figure 3*). Because each pixel has a set location in the original image, it is possible to convert the positions to new locations in a systematic manner.
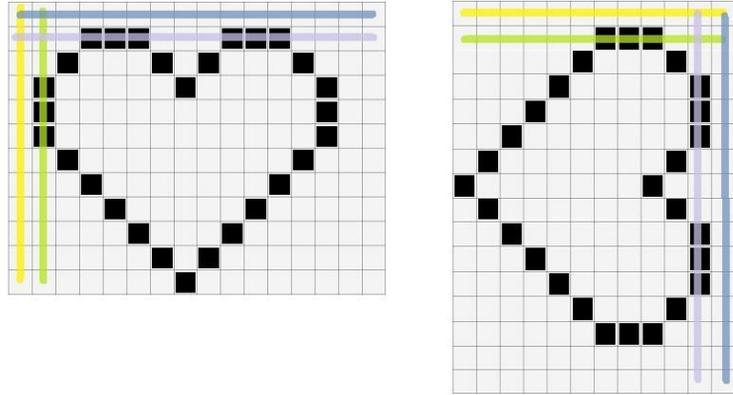

Figure 3: rotation clockwise with highlighted rows/columns

As well, it is easy to apply filters that can change the coloring of an image since the color value information is easily accessible and well organized. To achieve a black and white filter, each individual pixel has the average of its red, green, blue values taken; then the new pixel for the black and white image has the red, green, and blue values set to that average. Doing color swaps, the new pixel has the values for red and blue swapped from those in the original image.

Figure 4: red-blue swap

The representation of images using matrices has created a huge world of possibilities in digital image processing. The ones listed here are a slim picking of examples to demonstrate this. There are multitude more of kernels that can be applied, as well as applications of constants and changing of pixel positions (skewing, flipping, etc.). Matrix manipulation has made a world of difference in image processing.

Sources

Ludwig, Jamie (n.d.). *Image Convolution (http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf).* Portland State University.

Powell, Victor. Image Kernels Explained Visually (http://setosa.io/ev/image-kernels/).