

## Lab 11: Introduction to Arrays

### Background Exercises

For Part A and Part B of this lab, you may use your own jpeg image or you may use the image on Blackboard with a filename of Garden.JPG.

To import a jpeg file into MATLAB, first make sure the file is in your current directory. Use the *imread* command to import the file. For example, this is how Garden.JPG would be imported:

```
>> X = imread('Garden','jpg');
```

1. Import the Garden.JPG image or an image of your own using the imread command (don't leave off the semicolon at the end or you will see a display of every pixel in the image).
2. Look in your workspace window.

What size is X? \_\_\_\_\_

What variable type is X? \_\_\_\_\_

What is the range for this variable type? \_\_\_\_\_

3. View your image by executing this command: `>> imshow(X)`

4. Convert your image to grayscale and view it by executing the following commands:

```
>> Pic = 0.299*X(:,:,1) + 0.587*X(:,:,2) + 0.114*X(:,:,3);  
>> imshow(Pic)
```

What size is Pic? \_\_\_\_\_

What variable type is Pic? \_\_\_\_\_

Comment: Conversion to grayscale simply involves combining the red pane ( $X(:,:,1)$ ) and the green pane ( $X(:,:,2)$ ) and the blue pane ( $X(:,:,3)$ ) into a single matrix using weightings of 29.9% red, 58.7% green, and 11.4% blue.

5. Double click on Pic in the Workspace window to open it up in the Variable Editor window. Take a quick look at a couple of rows and columns of the values.

## A. Grayscale Image to Black and White Image

The grayscale image has 256 shades of gray ranging from black (pixel value of 0) to white (pixel value of 255). A pure black and white image would have only 2 colors: black and white so all the pixel values in the image would be either 0 (black) or 255 (white). To convert the image, we will choose a Threshold value between 0 and 255 and use the following relation:

$$\begin{aligned} \text{NewPic}(r,c) &= 255 && \text{if Pic}(r,c) \geq \text{Threshold} \\ \text{NewPic}(r,c) &= 0 && \text{if Pic}(r,c) < \text{Threshold} \end{aligned}$$

1. Download the template script file from Blackboard and re-name it as *Lab11A\_YourLastName*
2. Write a script file that does the following:
  - Prompts the user for the name of his/her jpg file using an input statement. Set up your input statement to expect a string.
  - Uses the *imread* command to read in the jpg image. Note: the filename will already be a string so don't use single quotes around it again.
  - Converts the color image to a grayscale image called Pic.
  - Prompts the user for a Threshold value from 0 to 255.
  - If the user enters an invalid value for Threshold, the script continues to prompt the user for a valid value until the user finally manages to follow directions.
  - Determine the size of Pic.
  - Create a new matrix called NewPic which is the same size as Pic, contains all zeros, and is type uint8. Note: everything in MATLAB defaults to a double. Use the uint8 function to convert to this variable type. Ex.  $M = \text{uint8}(M)$  will convert an array M to type uint8.
  - Use the conversion formula given above and the value of Threshold entered by the user to make NewPic a pure black and white image.
  - Use the *imshow* command to display the original grayscale image. Add a title indicating that this is the original image.
  - Use the command *figure* to open a new figure window then use the *imshow* command to display the black and white image. Add a title indicating that this is the Black and White image.
3. Check and make sure that your script only accepts valid values for Threshold from 0 to 255.
4. Run your script for three valid Threshold values and paste the results below along with your original grayscale image. Indicate what you chose for Threshold in each case.

### Paste Images Here:

5. Use your script to help answer the following questions:

What happens to the image as the Threshold value becomes close to zero? Why?

What happens to the image as the Threshold value becomes close to 255? Why?

### Part B Four Levels of Gray

In this part, we will allow for 4 different threshold values entered by the user in a vector. These four levels will determine the four levels of gray to appear in `NewPic`. Suppose the user enters the following: `Threshold = [50 100 150 200]`;

Then

$$\begin{aligned} \text{NewPic}(r,c) &= 50 && \text{if } \text{Pic}(r,c) \leq 50 \\ \text{NewPic}(r,c) &= 100 && \text{if } 50 < \text{Pic}(r,c) \leq 100 \\ \text{NewPic}(r,c) &= 150 && \text{if } 100 < \text{Pic}(r,c) \leq 150 \\ \text{NewPic}(r,c) &= 200 && \text{if } \text{Pic}(r,c) > 150 \end{aligned}$$

1. Download the template script file from Blackboard and re-name it as `Lab11B_YourLastName`
2. Write a script that does the following (you should be able to copy and paste some of your commands from Lab11A – just don't overwrite that old file):
  - Prompts the user for the name of his/her jpg file using an input statement. Set up your input statement to expect a string.
  - Uses the `imread` command to read in the jpg image. Note: the filename will already be a string so don't use single quotes around it again.
  - Converts the color image to a grayscale image called `Pic`.
  - Prompts the user to enter **a vector** of 4 Threshold values ranging from 0 to 255. **This should be a single input statement where the user is expected to enter four values enclosed in square brackets.**
  - If the user enters an invalid value in the Threshold vector or the Threshold vector doesn't have exactly four values, the script continues to prompt the user for a valid values and valid length until the user finally manages to follow directions.

**Hint:** Try the following commands in MATLAB and see if you can make use of this:

```
>> Threshold = [ -50 -100 200 300];  
>> sum(Threshold < 0)
```

```
>> Threshold = [ -50 -100 200 300];  
>> sum(Threshold > 255)
```

- The user could potentially enter 4 valid values which aren't in numerical order. Use the `sort` command in MATLAB to sort the values from smallest to largest:

```
Threshold = sort(Threshold);
```

- Determine the size of `Pic`.
- Create a new matrix called `NewPic` which is the same size as `Pic`, contains all zeros, and is type `uint8`.
- Use the conversion formula described above and the values of `Threshold` entered by the user to make `NewPic`.