Republic of the Philippines
**CAVITE STATE UNIVERSITY**
**Don Severino delas Alas Campus**
Indang, Cavite

# ELECTIVE 3

Midterm Exam
**Image Processing in Octave**

Score

*Submitted by:*
**Pacomio, John Emmanuel C.**
**Thursday 3:00pm – 4:00pm / BSCpE 4-2**

*Date Submitted*
**24-11-2022**

*Submitted to:*

**Engr. Maria Rizette H. Sayo**

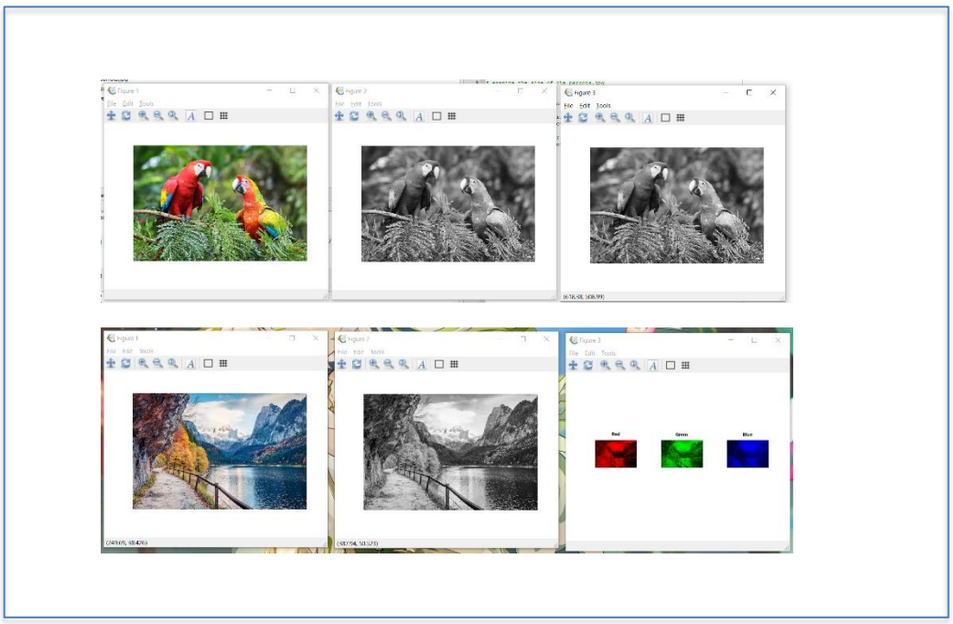
Methodology

I. Importing, Displaying, and Converting Images
   a. Using the Editor of Octave, create a program that will load and display parrots.jpg
   b. Examine the size of the parrots by typing whos to find out the size of the image that you have read in
   c. Convert the class uint8 color image parrots to a gray scale image, and display the full intensity range gray-scale image using the imshow command
   d. Covert the true color image to a gray-scale image
   e. Save the program to this format parrots.m

II. Display of Color Images
   a. Open the image file nature.jpg from the source folder
   b. Read in the file nature.jpg and display it on the screen as a reference image How large an image is created when we use the RGB representation compared to a gray-scale image conversion of it?
   c. Assign an image color that intensifies red, green, and blue and display each image in one window
   d. Convert each image file extension to png
   e. Save the program as nature.jpg

*Note: Take a screen shot of the output display of the converted images and do not forget to write your Octave code to this manuscript*

Screen shot of Output Image



Source Codes:

```
clc;
clf;
close all;
clear all;
# Load and display parrots.jpg
parrot = imread('D:/College Files/CPEN111/MIDTERM\parrots.jpg');
figure(1), imshow(parrot);

# examine the size of the parrots.jpg
whos parrot

# Convert the class uint8 color image parrots to a gray scale
image,
# and display the full intensity range gray-scale image using the
imshow command
parrot_g = rgb2gray(parrot);
figure(2), imshow(parrot_g, [0 255]);

# Covert the true color image to a gray-scale image
parrot_gray = rgb2gray(parrot);
figure(3), imshow(parrot_gray);
```

```
clc;
clf;
close all;
clear all;

# Open the image file nature.jpg from the source folder
source = imread('D:/College Files/CPEN111/MIDTERM/nature.jpg');
figure(1), imshow(source);

# comparing image size of RGB nature.jpg to grayscale
whos source
grayscale = rgb2gray(source);
figure(2), imshow(grayscale);
imwrite(grayscale, 'D:/College
Files/CPEN111/MIDTERM/grayscale.png');
whos grayscale

# intensifies red then save it
img_red = source;
img_red(:,:,2) = 0;
img_red(:,:,3) = 0;
imwrite(img_red, 'D:/College
Files/CPEN111/MIDTERM/red_nature.png');

# intensifies green then save it
img_green = source;
img_green(:,:,1) = 0;
img_green(:,:,3) = 0;
imwrite(img_green, 'D:/College
Files/CPEN111/MIDTERM/green_nature.png');

# intensifies blue then save it
img_blue = source;
img_blue(:,:,1) = 0;
img_blue(:,:,2) = 0;
imwrite(img_blue, 'D:/College
Files/CPEN111/MIDTERM/blue_nature.png');

# displaying intensified red, green, and blue
figure(3),
subplot(1,3,1), imshow(img_red), title('Red');
subplot(1,3,2), imshow(img_green), title('Green');
subplot(1,3,3), imshow(img_blue), title('Blue');
```

## Conclusion

I was able to convert a true image to a grayscale image for this midterm exam using the Octave's rgb2gray function in the first methodology. I was able to manipulate the color intensity of the source image while using the second methodology. After conducting the examination, we can now manipulate images, which we could use to try to produce data in the future for AI to learn.

Rubrics in Grading the Midterm Exam

| | A – Excellent | B – Good | C – Fair | D – Needs Improvement |
|---|---|---|---|---|
| Specifications | The Program works and meets all of the specifications | The program works and produces the correct results and displays them correctly. It also meets the most of the other specifications | The program produces correct results but does not display them correctly | The program is producing incorrect results |
| Readability | The code is exceptionally well organized and very easy to follow | The code is fairly easy to read | The code is readable only by someone who knows what it is supposed to be doing | The code is poorly organized and very difficult to read. |
| Reusability | The code could be reused as a whole or each routine could be reused | Most of the code could be reused in other programs | Some parts of the code could be reused in other programs | The code is not organized for reusability |
| Documentation | The documentation is well written and clearly explains what the code is accomplishing and how | The documentation consists of embedded comment and some simple header documentation that is somewhat useful | The documentation is simply comments embedded in the code with some simple header comments separating routines | The documentation is simply comments embedded in the code and does not help the reader understand the codes |
| Efficiency | The code is extremely efficient without sacrificing readability and understanding. | The code is fairly efficient without sacrificing readability and understanding | The code is brute force and unnecessarily long | The code is huge and appears to be patched together |
| TOTAL | | | | |