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:*
**Aplacador, Bryan Austin D.**
**Thursday 3-4:30 / 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

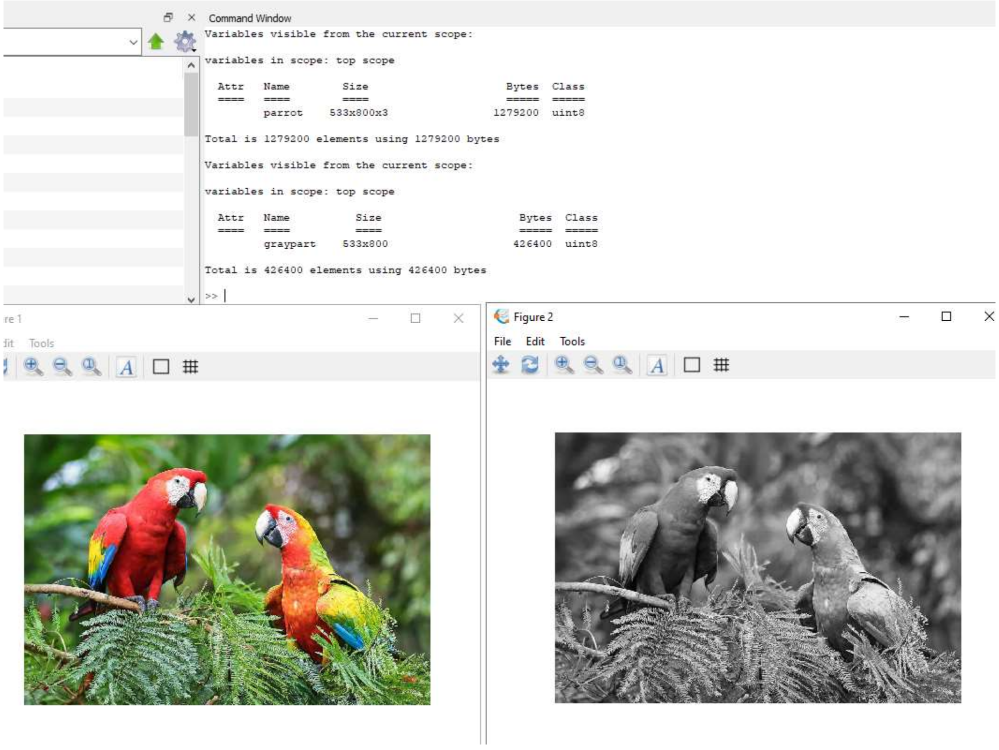
II. Display of Color Images

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*

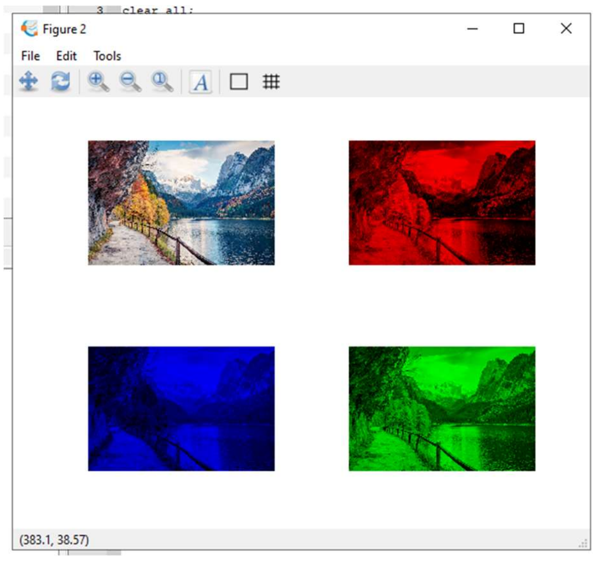
Screen shot of Output Image

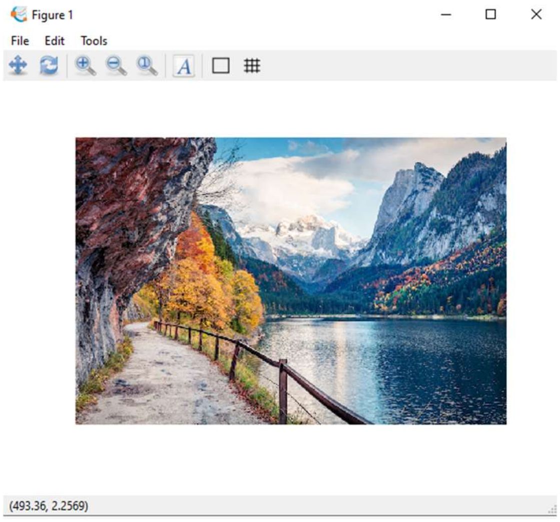
I. Importing, Displaying and Converting Images



```
Variables visible from the current scope:

variables in scope: top scope

  Attr   Name        Size                     Bytes  Class
  ====   ====        ====                     =====  =====
         parrot    533x800x3                1279200  uint8

Total is 1279200 elements using 1279200 bytes

Variables visible from the current scope:

variables in scope: top scope

  Attr   Name          Size                     Bytes  Class
  ====   ====          ====                     =====  =====
         graypart    533x800                   426400  uint8

Total is 426400 elements using 426400 bytes

>> 
```

II.





Source Codes:

```
I.
clc;
clf;
clear all;
close all;
pkg load image;

parrot = imread('C:\Users\Zchemisc\Desktop\parrots.jpg');
figure(1),imshow(parrot)
whos parrot


graypart = rgb2gray(parrot);
figure(2),imshow(graypart)
whos graypart
```

II.

```
clc;
clf;
clear all;
close all;
pkg load image;


nature = imread('C:\Users\Zchemisc\Desktop\nature.jpg');
figure(1),imshow(nature);

blue = nature;
blue(:,:,1)=0;
blue(:,:,2)=0;

green = nature;
green(:,:,1)=0;
green(:,:,3)=0;

red = nature;
```

```
red(:,:,3)=0;
red(:,:,2)=0;

%subplot(2,2,1),imshow(nature);
%subplot(2,2,2),imshow(red);
%subplot(2,2,3),imshow(blue);
%subplot(2,2,4),imshow(green);

imwrite(nature,'C:\Users\Zchemisc\Desktop\nature.png')
imwrite(blue,'C:\Users\Zchemisc\Desktop\nature_blue.jpg')
imwrite(green,'C:\Users\Zchemisc\Desktop\nature_green.jpg')
imwrite(red,'C:\Users\Zchemisc\Desktop\nature_red.jpg')
```

## Conclusion

In the first part, which is Importing, Displaying, and Converting Images, to read I used the imread function, and to display I used the imshow function. This whos function shows the information of the image name, size, class and bytes. I used rgb2gray() to convert the true color image to a grayscale image for the grayscale.

In the second part, "Display of Color Images," the image will turn red, blue, and green as it is expressed using this (:,:,2). To convert the extension file, I used the imwrite function. So, each image that is displayed in color will be saved and converted to a png extension file.

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 | | | | |

https://github.com/AplacadorBryanAustin/CPEN111.git