ELECTIVE 3

Midterm Exam
Image Processing in Octave
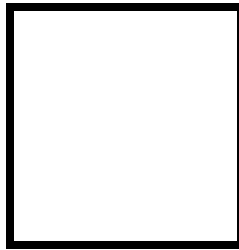
Score

Submitted by:
Gandingco, Aira J.
CPEN111 / 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 **(PARROTS)**
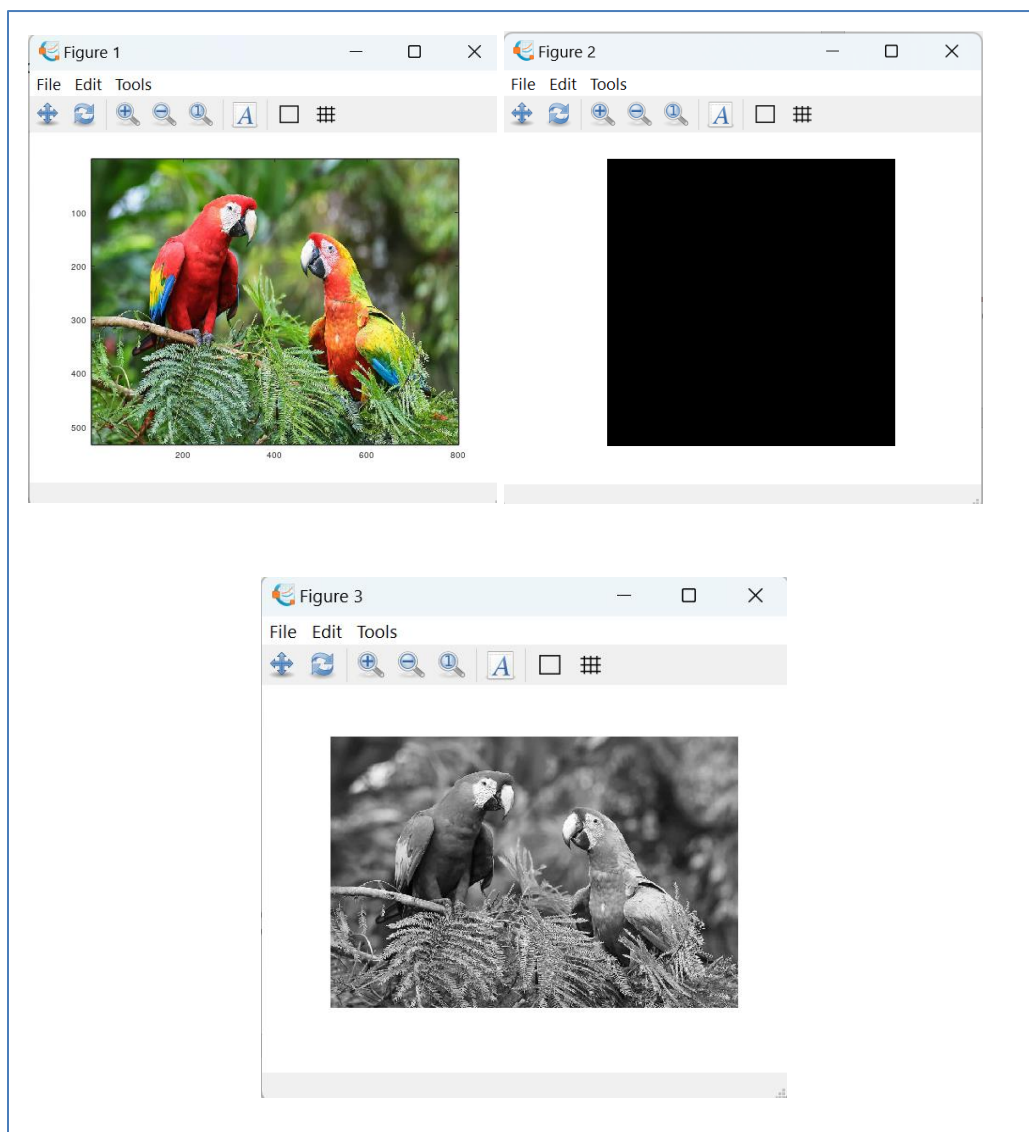
Source Codes:

```
%Gandingco, Aira J
%BSCPE 4-2
%Midterms

clc;
clf;
close all;
clear all;

%a program that will load and display parrots.jpg
parrots = imread('C:\Users\User\Desktop\4TH YEAR\CPEN
111/parrots.jpg');
figure, imshow(parrots);

%b Examine the size of the parrots by typing whos to find out the size of the
image that you have read in
whos parrots;

%%%cConvert the class uint8 color image parrots to a gray scale image, and
display the full intensity range gray-scale image using the imshow command
%%%dCovert the true color image to a gray-scale image
unit8_gray=imagesc(parrots, [0, 255]); colormap(gray);
figure(2), imshow(unit8_gray);
whos unit8_gray;

#Convert the true color image to a gray-scale image
gray_parrots = rgb2gray(parrots);
figure(3), imshow(gray_parrots);
```
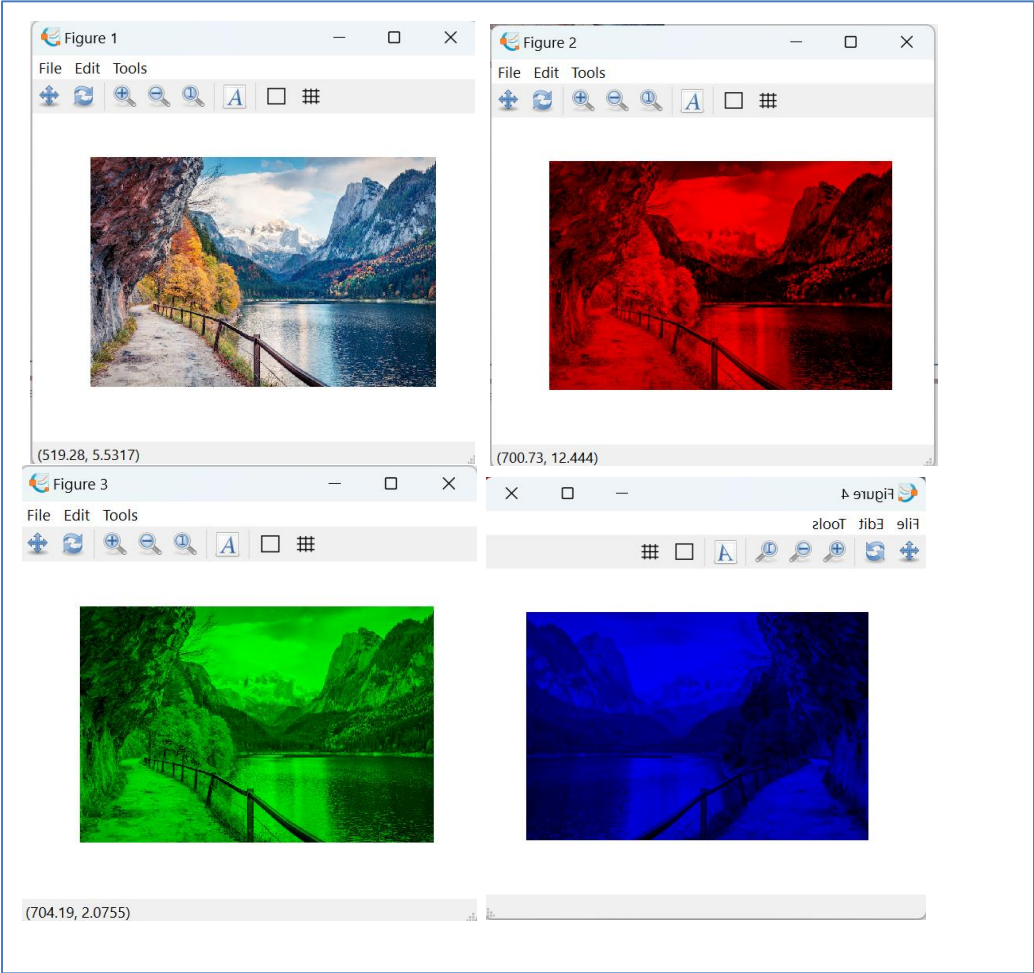
Screen shot of Output Image **(NATURE)**

Source Codes:

```
%Gandingco, Aira J
%BSCPE 4-2
%Midterms

clc;
clf;
clear all;
close all;

% a.    Open the image file nature.jpg
%b.     Read in the file nature.jpg and display it on the screen as a reference
image
ref_nature = imread('C:\Users\User\Desktop\4TH YEAR\CPEN
111/nature.jpg');
figure, imshow(ref_nature);

whos ref_naturenature;

gray_nature = rgb2gray(ref_nature);

%c.     Assign an image color that intensifies red, green, and blue and
display each image in one window
whos gray_nature;

red_img = ref_nature;
red_img (:,:,2)=0;
red_img (:,:,3)=0;
figure(2), imshow(red_img)
```

Conclusion

      I discovered that the Octave has routines for processing photos through this exercise. I applied my understanding of how to import, display, convert photos using the imshow command, and even display color images using Octave, to the provided challenge. It's crucial to discern between grayscale and RGB photos. Since less information must be provided for each pixel, these images differ from all other types of color images. The rgb2gray(RGB) command transforms the truecolor RGB image into an image in grayscale. By removing the hue and saturation data while keeping the luminance, the rgb2gray function converts RGB images to grayscale.

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	Most of the code could be reused in other programs	Some parts of the code could be	The code is not organized for reusability

	each routine could be reused		reused in other programs	
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				