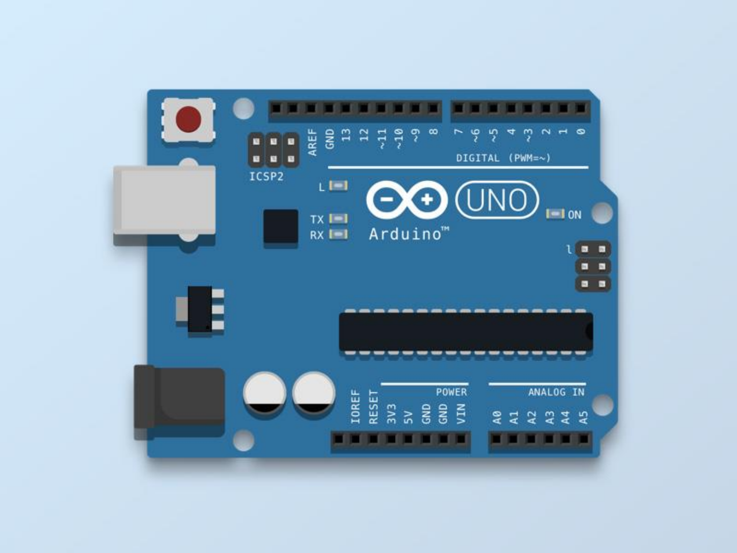
Color Recognition with Matlab Image Processing and Matlab Interfacing with Arduino

SAYAN SETH

Indian Institute of Engineering Science and Technology, Shibpur

HARDWARES USED:

1. Arduino Uno R3
2. LED x 3
3. USB Webcam

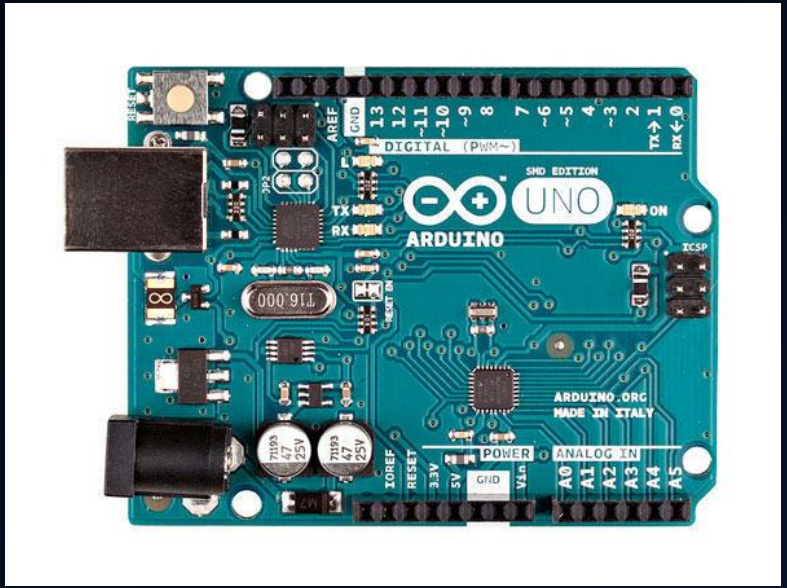


SOFTWARES USED:

1. Arduino IDE
2. Matlab R2017a
3. Fritzing

Arduino Uno R3

Arduino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP(In Circuit Serial Programmer) header and a reset button.

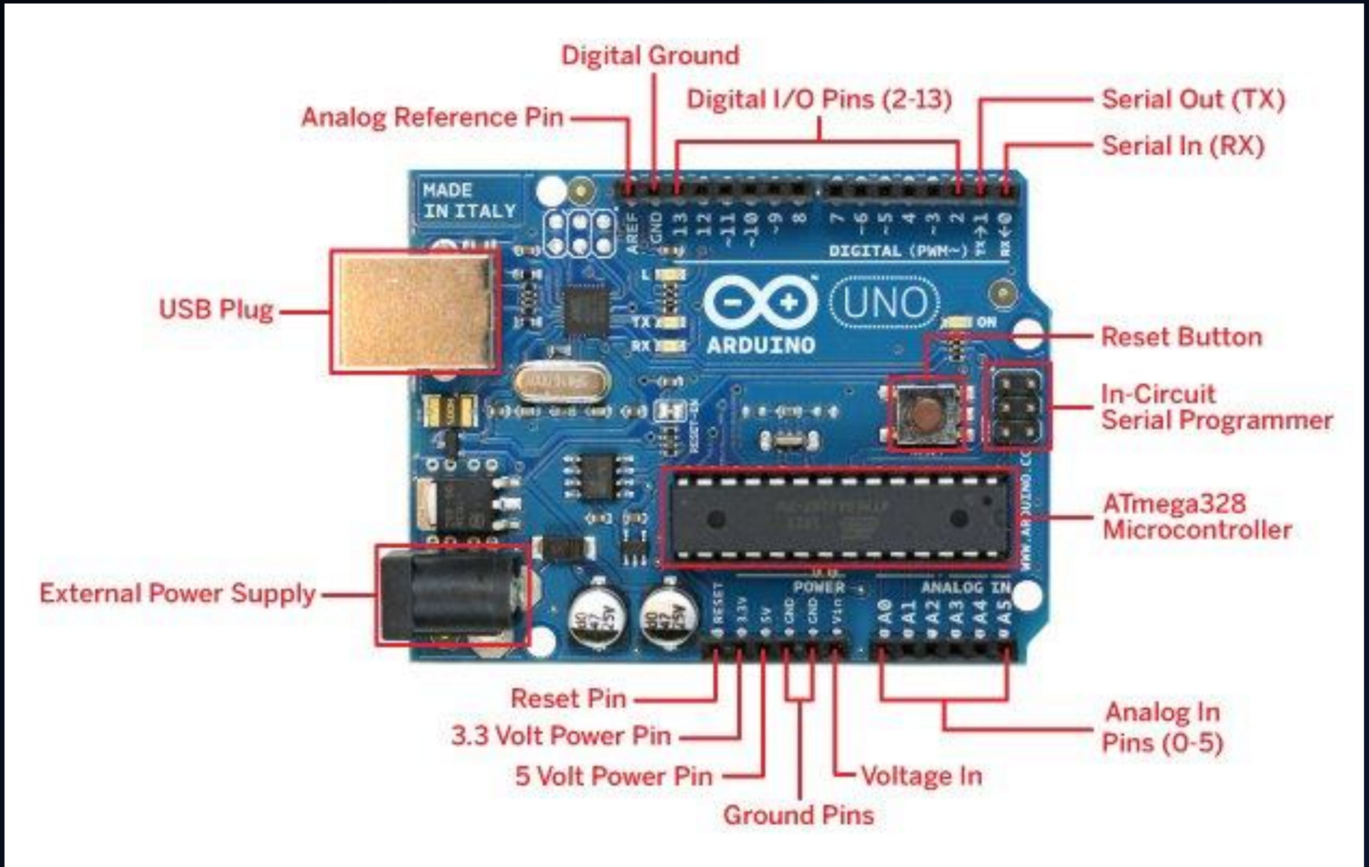


Features

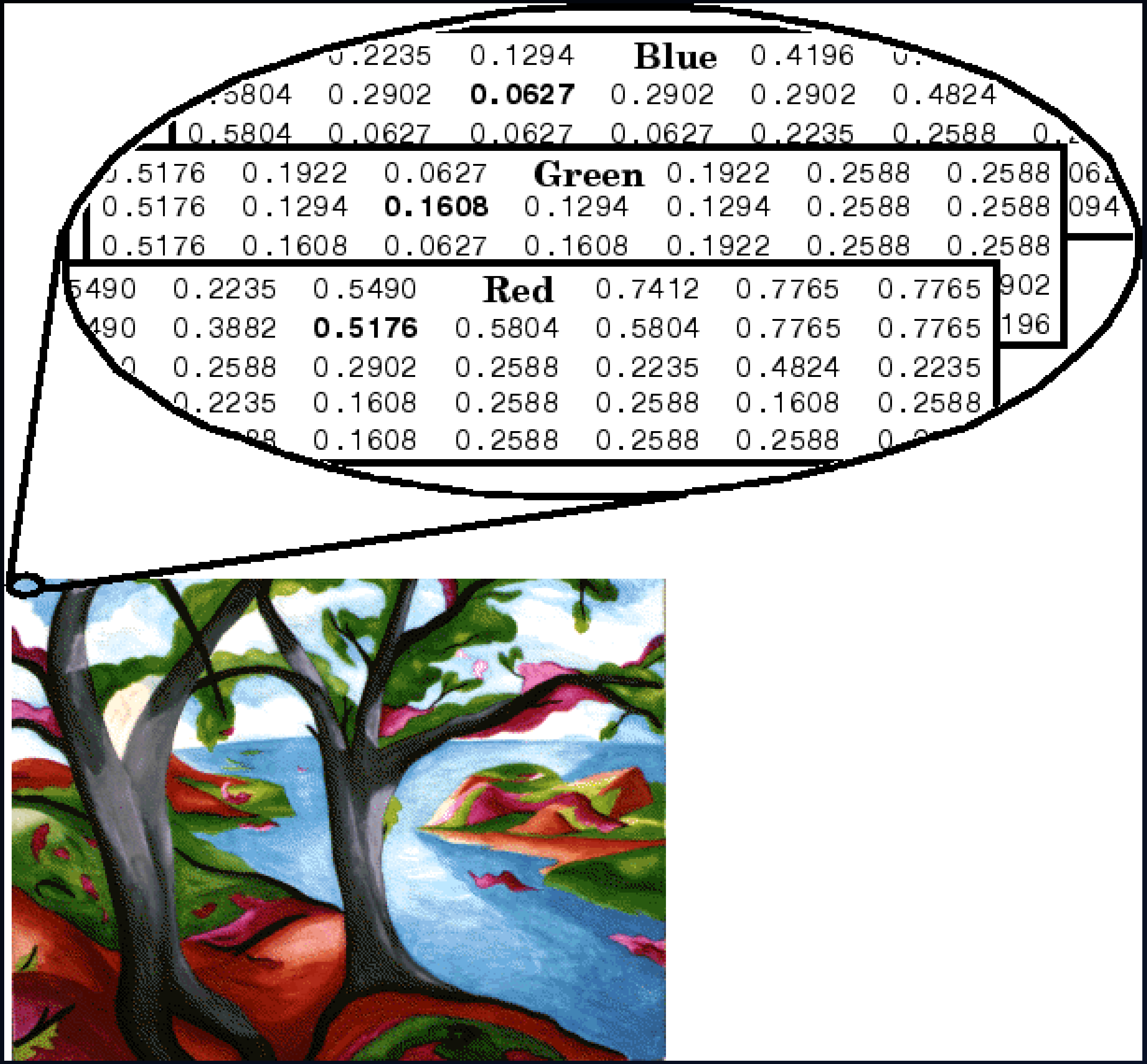
- ATmega328 microcontroller
- Input voltage - 7-12V
- 14 Digital I/O Pins (6 PWM outputs)
- 6 Analog Inputs
- 32k Flash Memory
- 16Mhz Clock Speed
- SRAM 2 KB
- EEPROM 1KB

DIFFERENT PARTS OF ARDUINO UNO

Serial: 0 (RX) and 1 (TX): Used to receive (RX) and transmit (TX) TTL serial data.
PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
Analog Pins: It support 10-bit Analog -to-digital conversion (ADC) using the analogRead() function.



RGB Images



.2235 0.1294 Blue 0.4196 0.
.5804 0.2902 **0.0627** 0.2902 0.2902 0.4824
0.5804 0.0627 0.0627 0.0627 0.2235 0.2588 0.

.5176 0.1922 0.0627 Green 0.1922 0.2588 0.2588 062
0.5176 0.1294 **0.1608** 0.1294 0.1294 0.2588 0.2588 094
0.5176 0.1608 0.0627 0.1608 0.1922 0.2588 0.2588

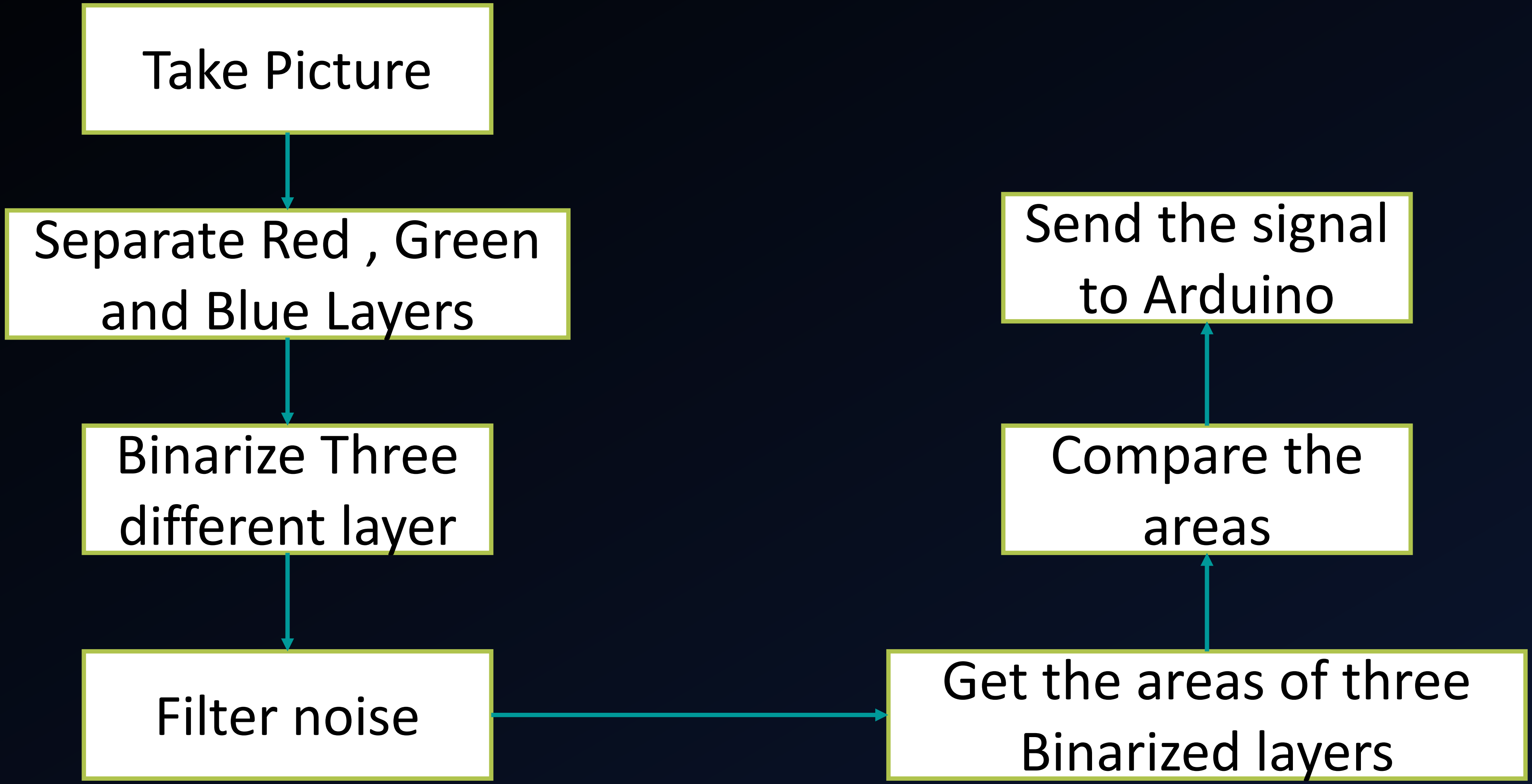
5490 0.2235 0.5490 Red 0.7412 0.7765 0.7765 902
490 0.3882 **0.5176** 0.5804 0.5804 0.7765 0.7765 196
0 0.2588 0.2902 0.2588 0.2235 0.4824 0.2235
.2235 0.1608 0.2588 0.2588 0.1608 0.2588
88 0.1608 0.2588 0.2588 0.2588 0.

The color components of an 8-bit RGB image are integers in the range [0, 255].

A pixel whose color components are (255,255,255) is displayed as White.

The image command displays an RGB image correctly whether its class is double, uint8, or uint16.

Image Processing Algorithm



Take Picture

Separate Red , Green and Blue Layers

Binarize Three different layer

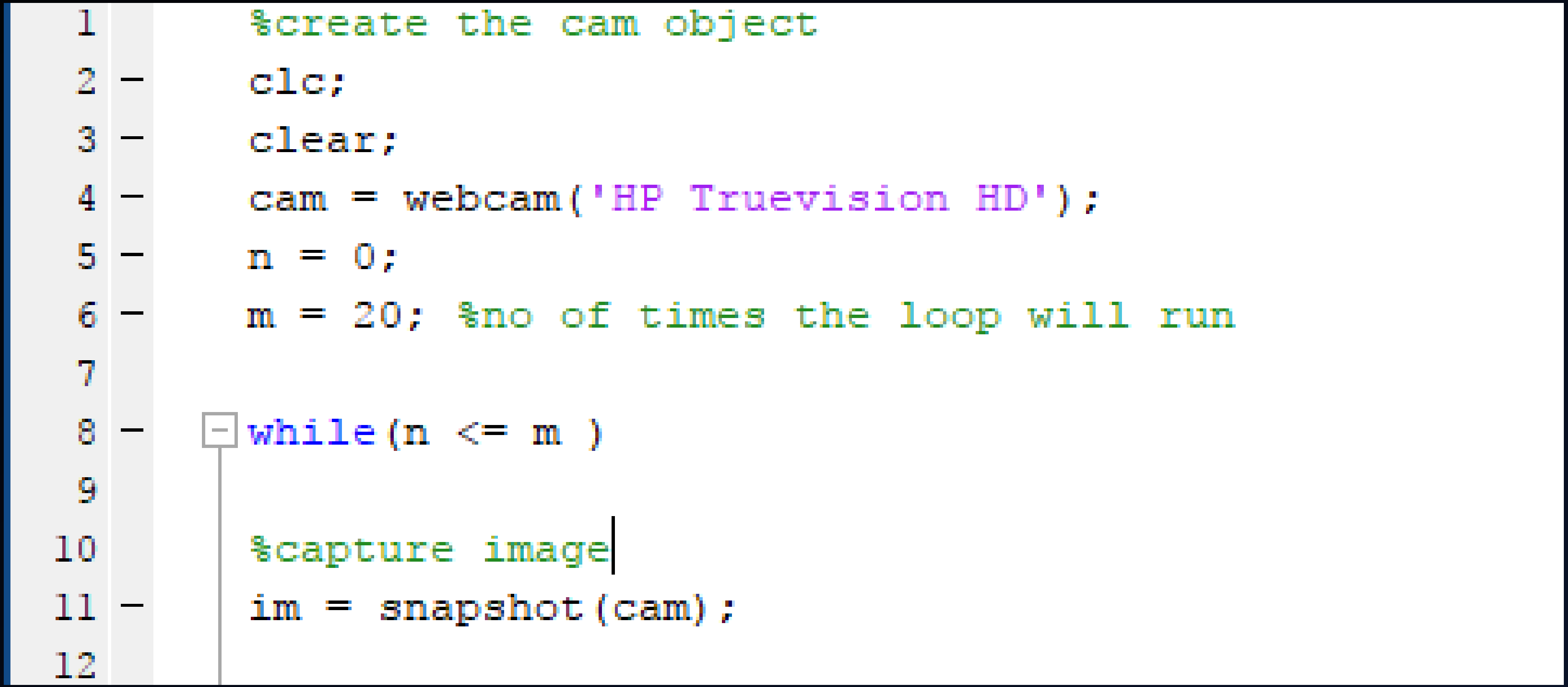
Filter noise

Get the areas of three Binarized layers

Compare the areas

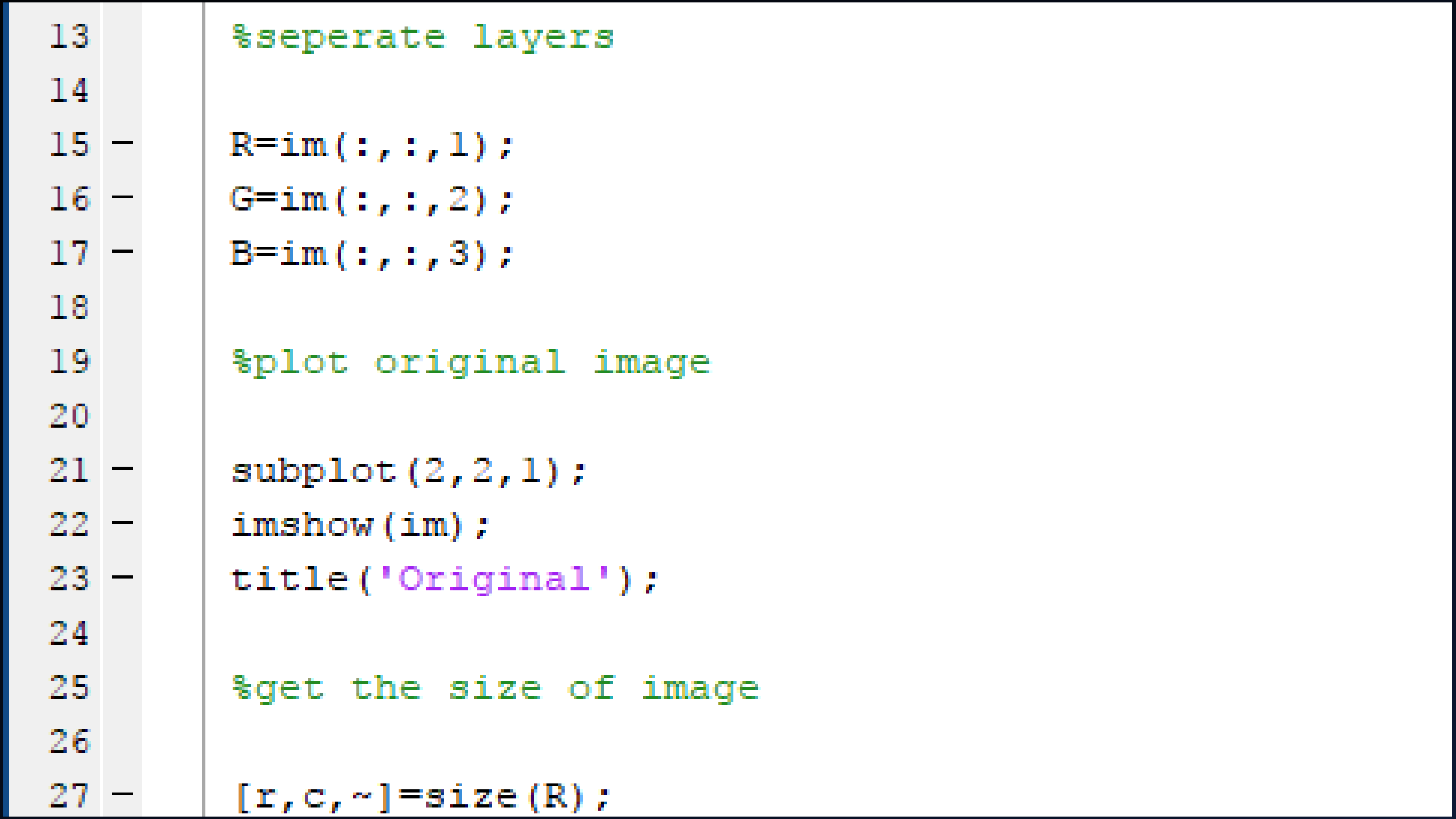
Send the signal to Arduino

Matlab Code



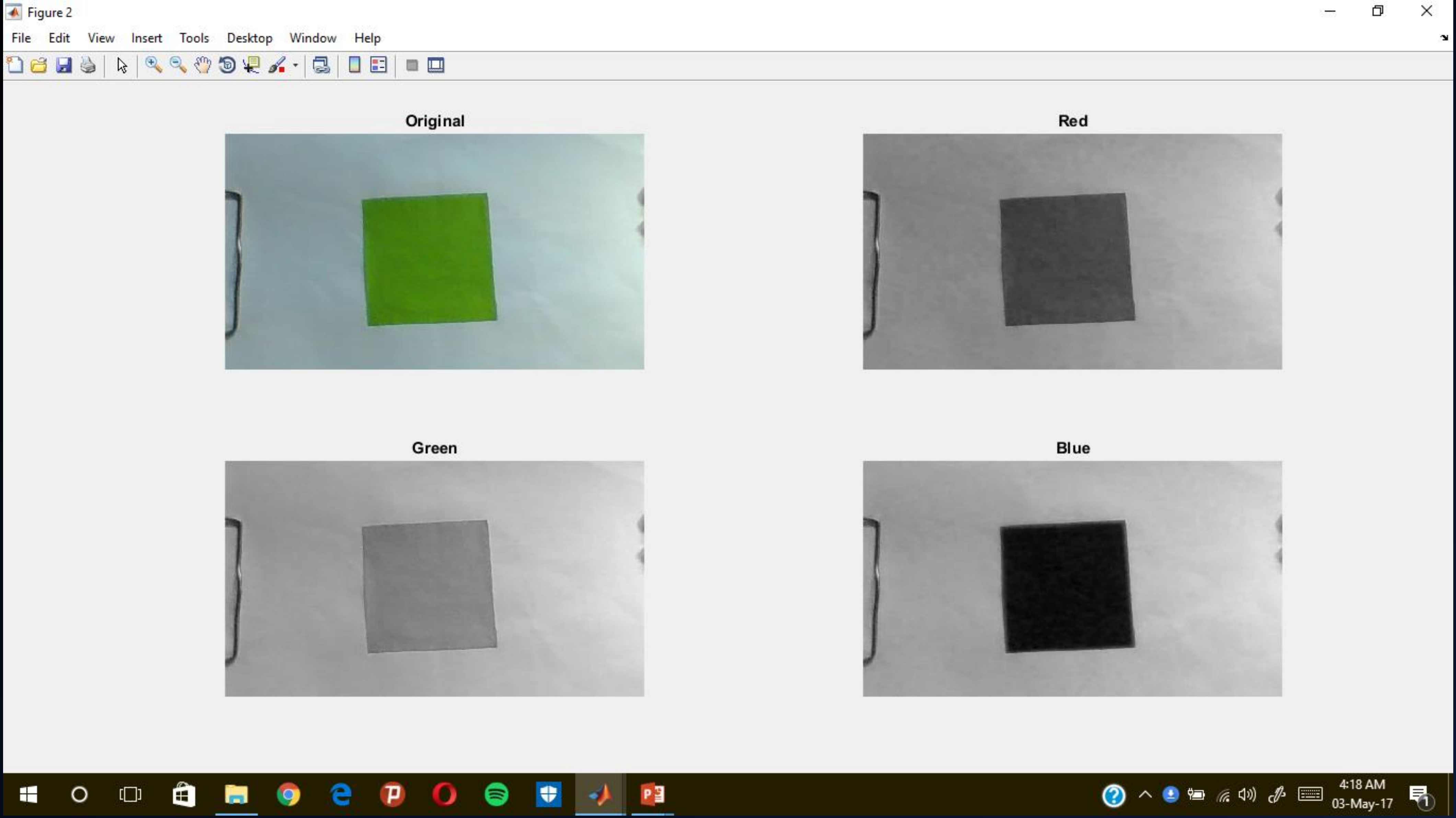
```
%create the cam object
clc;
clear;
cam = webcam('HP Truevision HD');
n = 0;
m = 20; %no of times the loop will run

while(n <= m )

%capture image
im = snapshot(cam);

```

Matlab Code

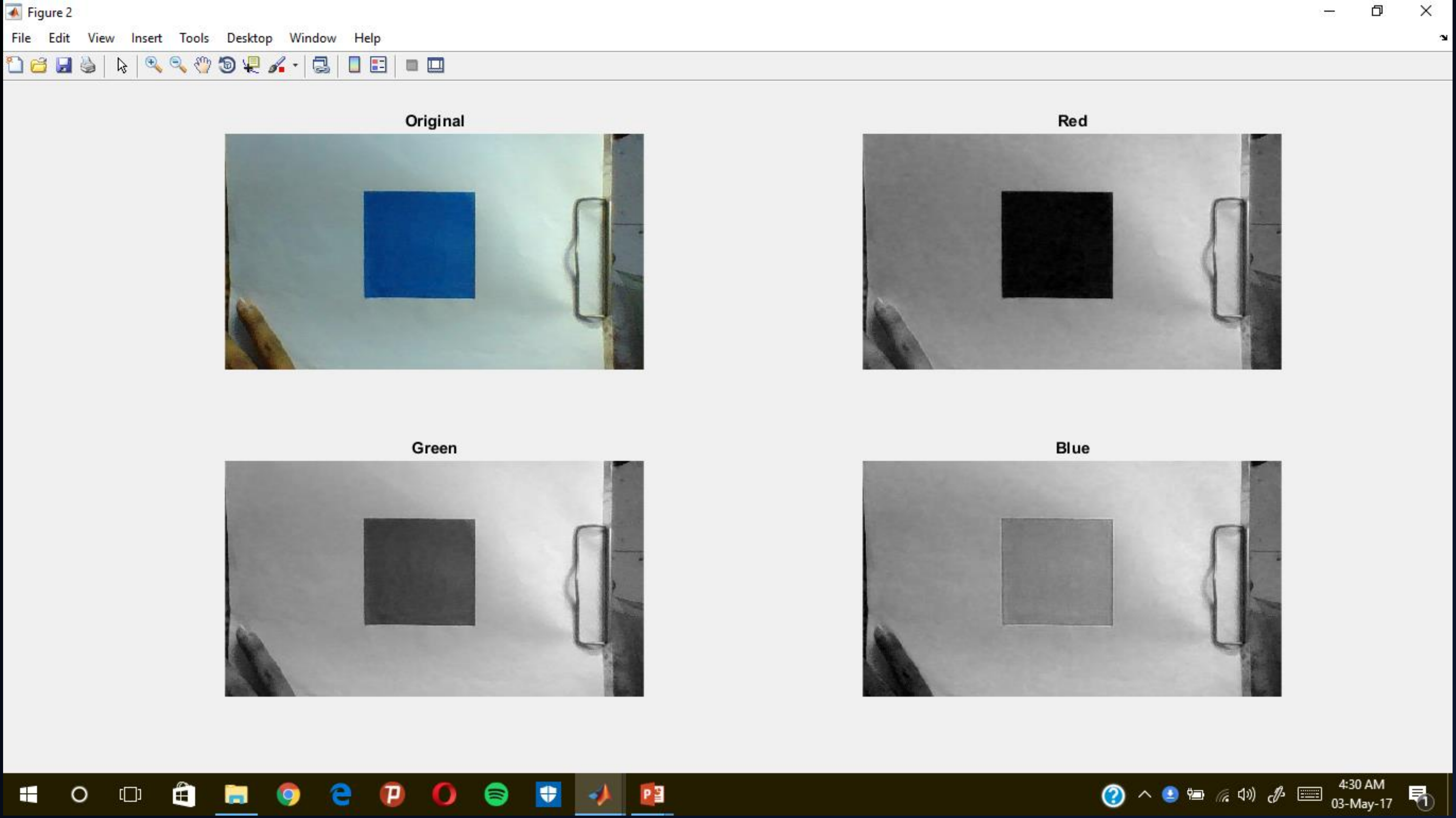


```
%seperate layers

R=im(:,:,1);
G=im(:,:,2);
B=im(:,:,3);

%plot original image

subplot(2,2,1);
imshow(im);
title('Original');

%get the size of image

[r,c,~]=size(R);
```

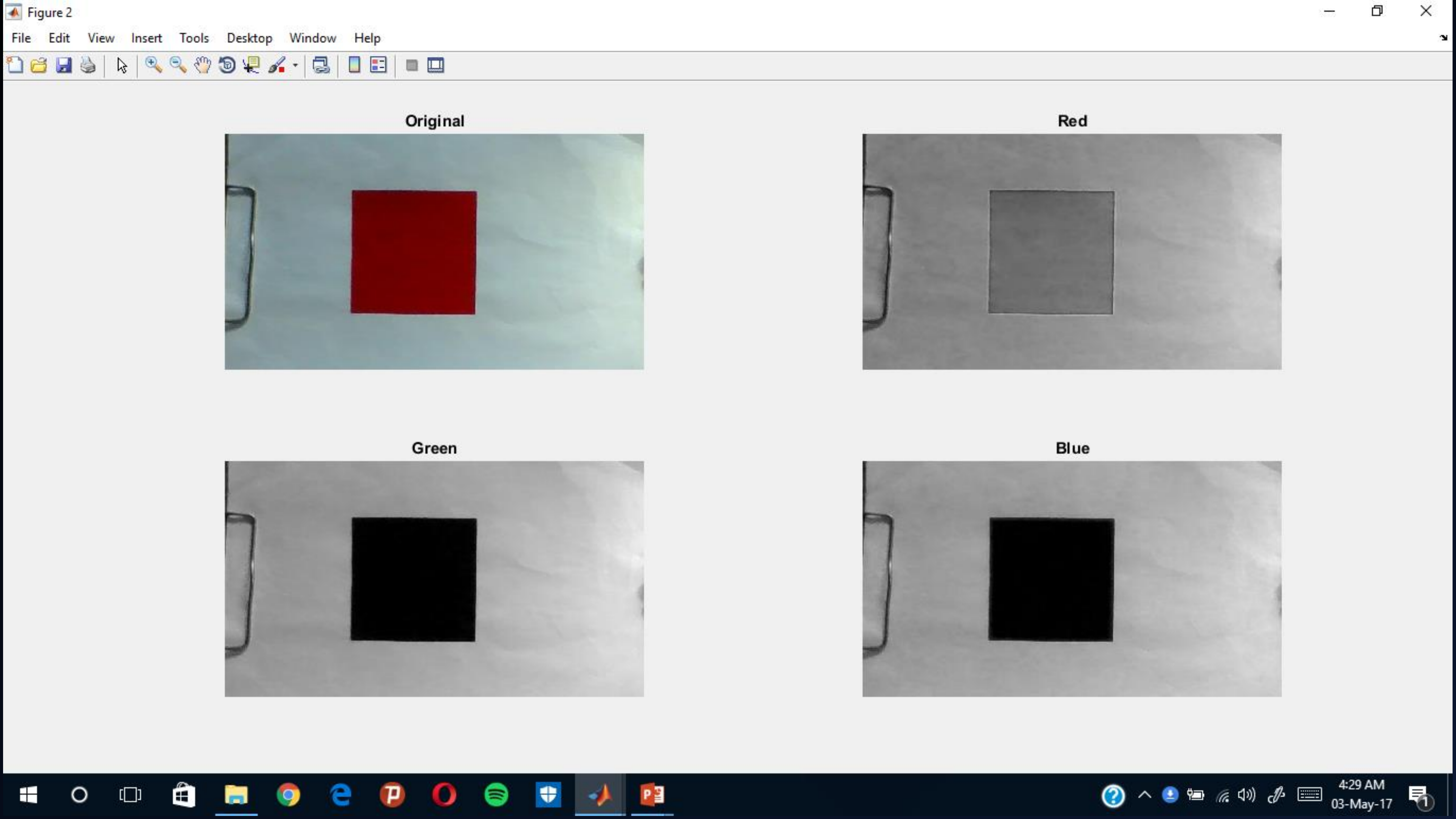
Single Layer Image (Green)



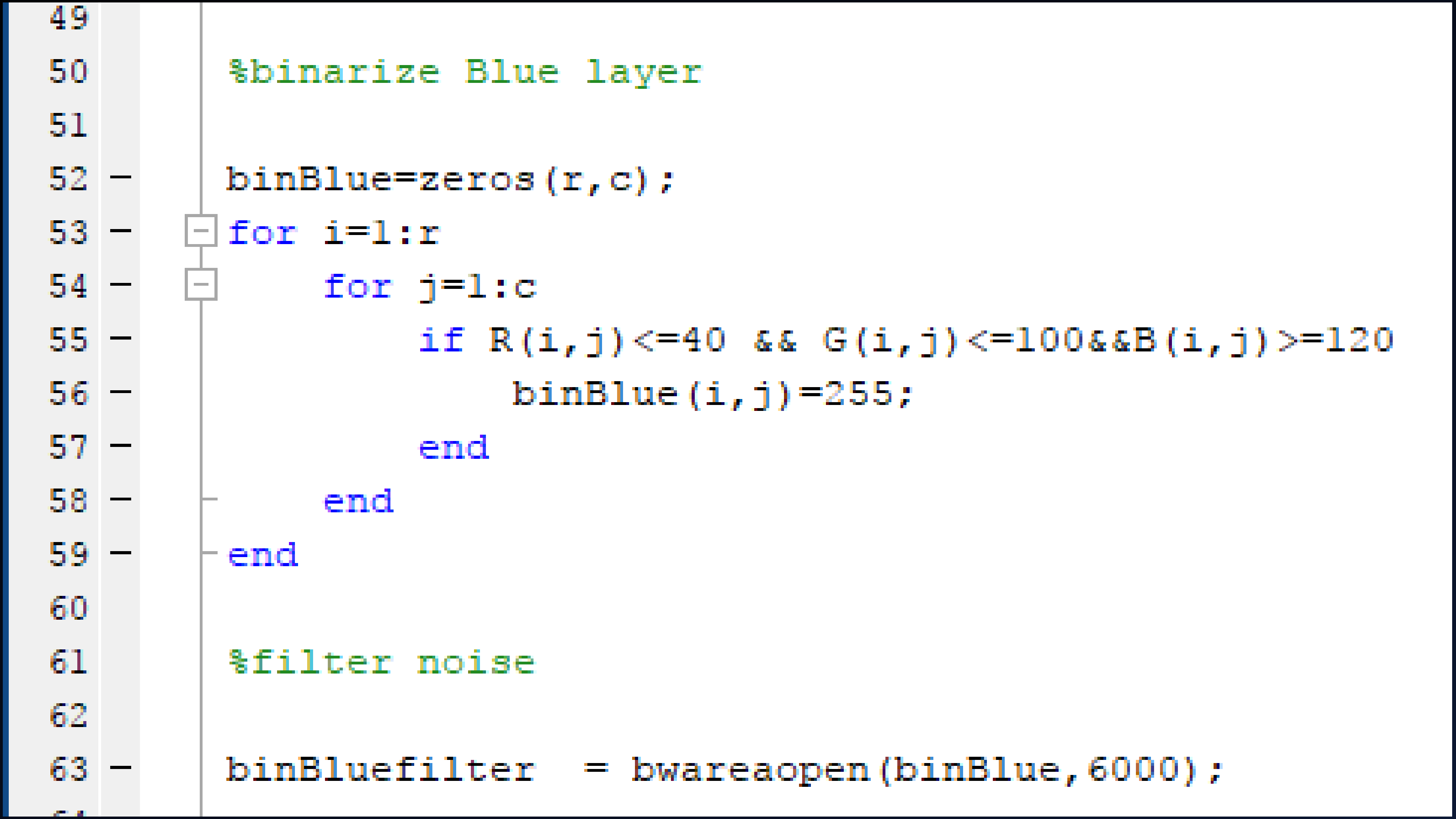
Single Layer Image (Blue)



Single Layer Image (Red)

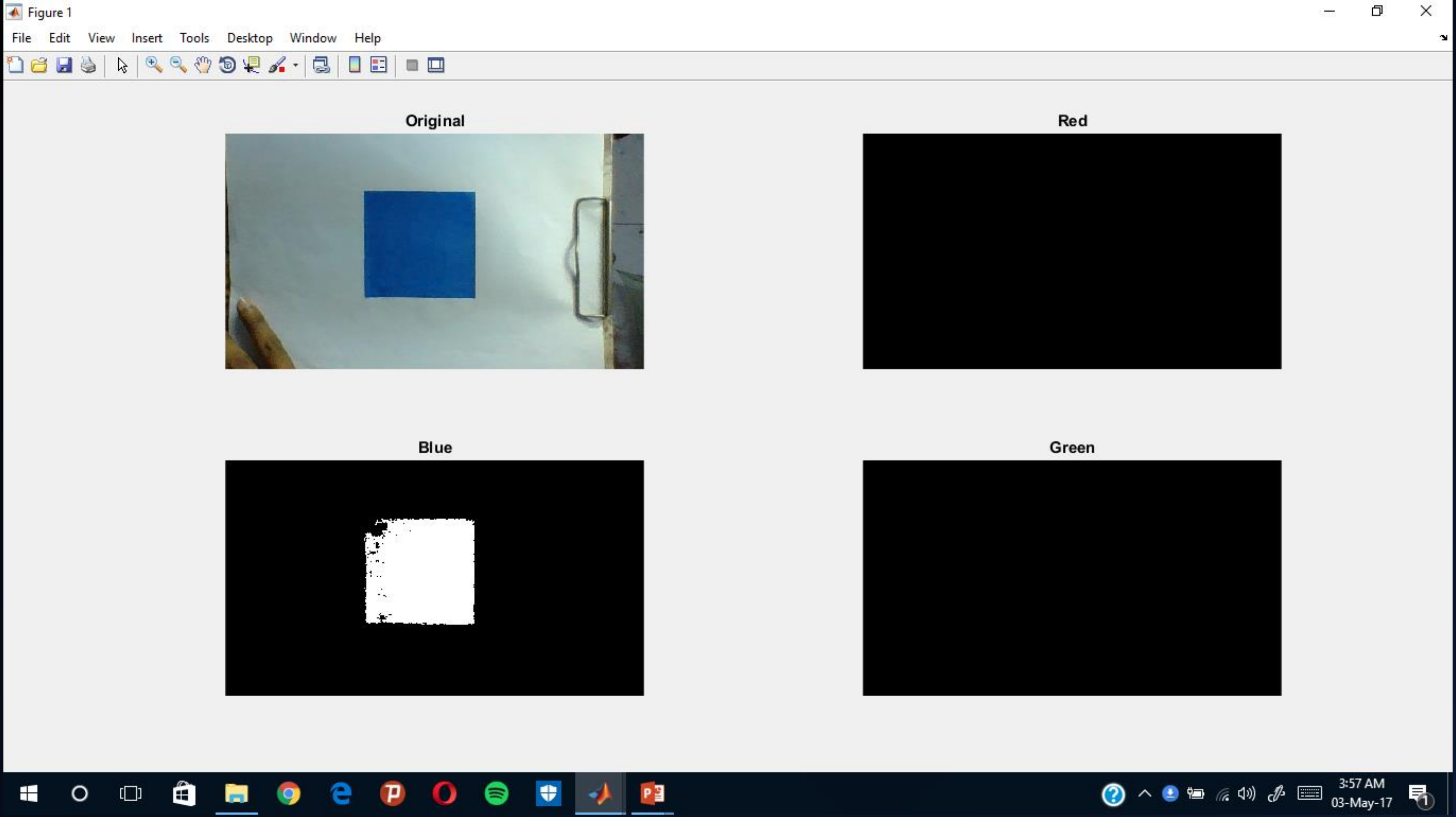


Matlab Code

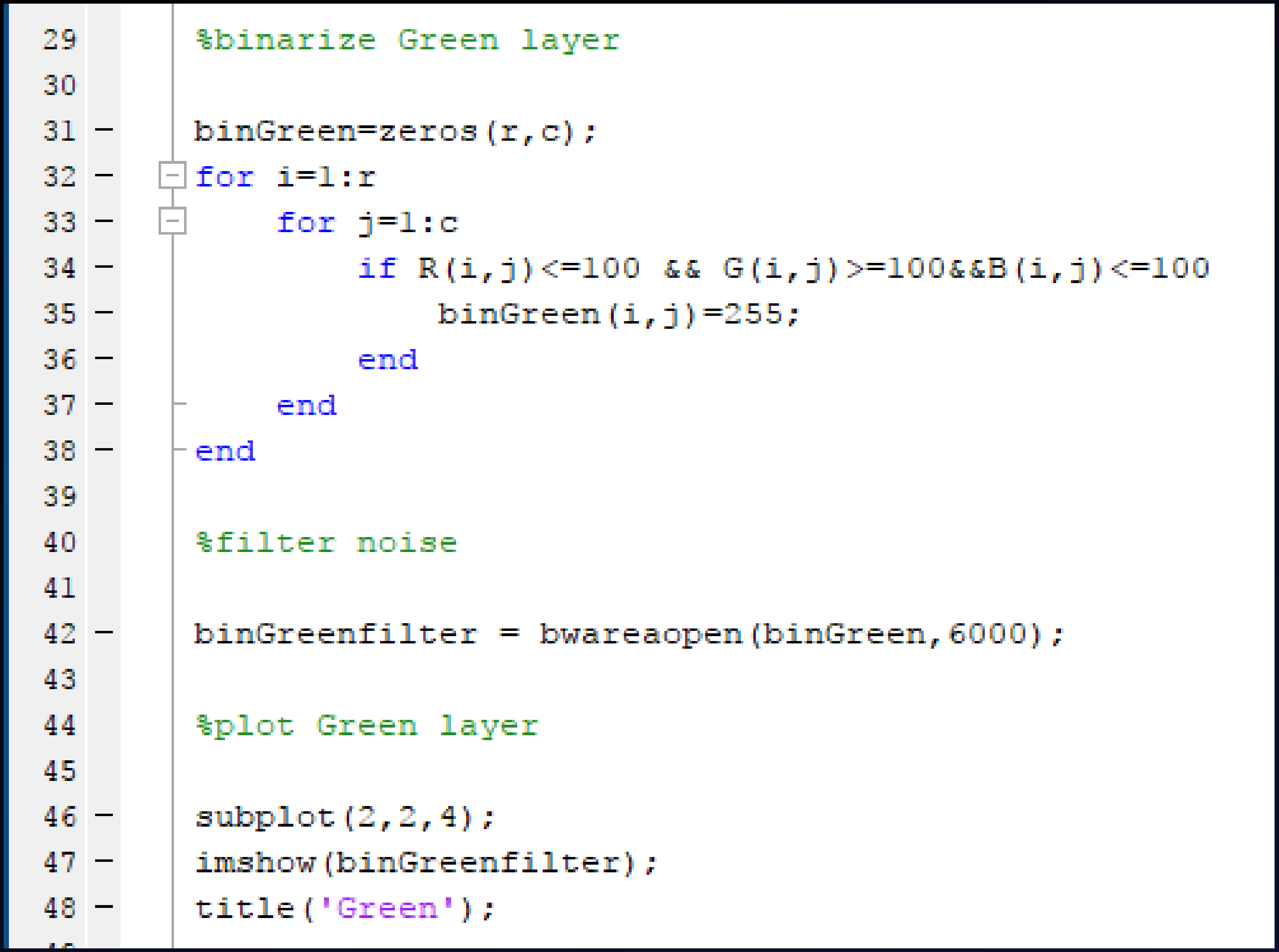


```matlab

%binarize Blue layer

binBlue=zeros(r,c);
for i=1:r
    for j=1:c
        if R(i,j)<=40 && G(i,j)<=100&&B(i,j)>=120
            binBlue(i,j)=255;
        end
    end
end

%filter noise

binBluefilter  = bwareaopen(binBlue,6000);
```

Binarized Image: Blue

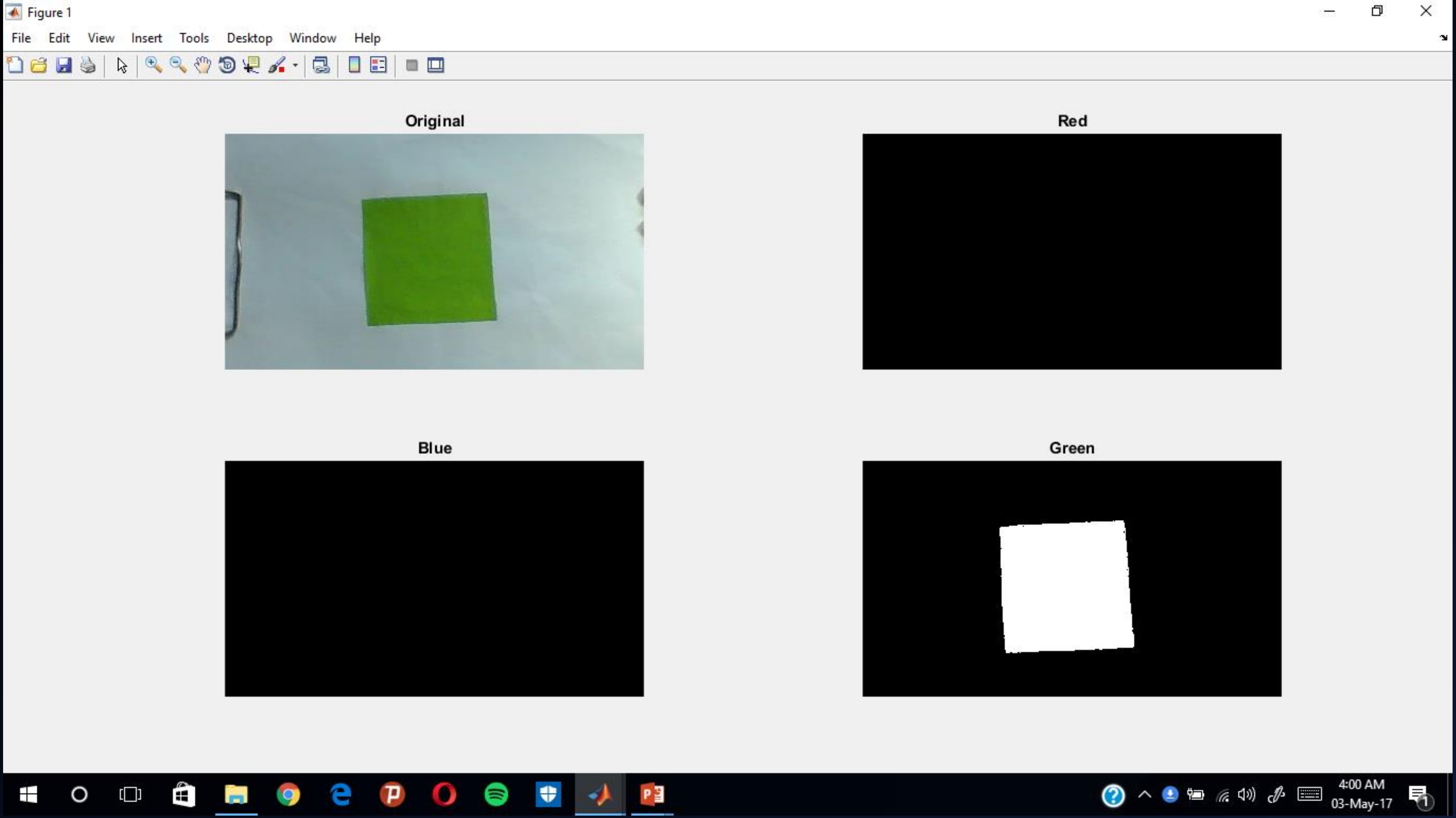


Matlab Code

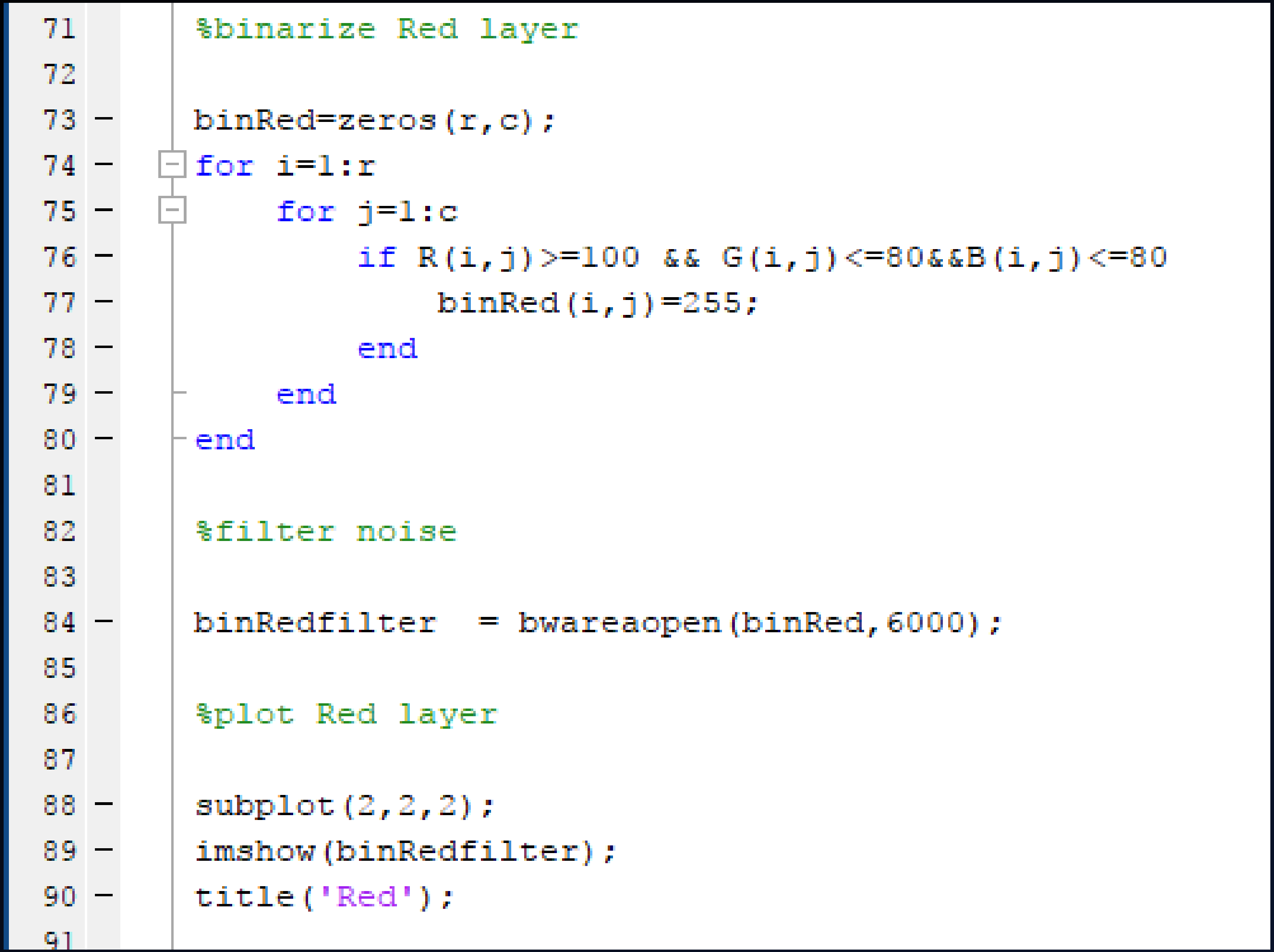


```
%binarize Green layer

binGreen=zeros(r,c);
for i=1:r
    for j=1:c
        if R(i,j)<=100 && G(i,j)>=100&&B(i,j)<=100
            binGreen(i,j)=255;
        end
    end
end

%filter noise

binGreenfilter = bwareaopen(binGreen,6000);

%plot Green layer

subplot(2,2,4);
imshow(binGreenfilter);
title('Green');
```

Binarized Image: Green

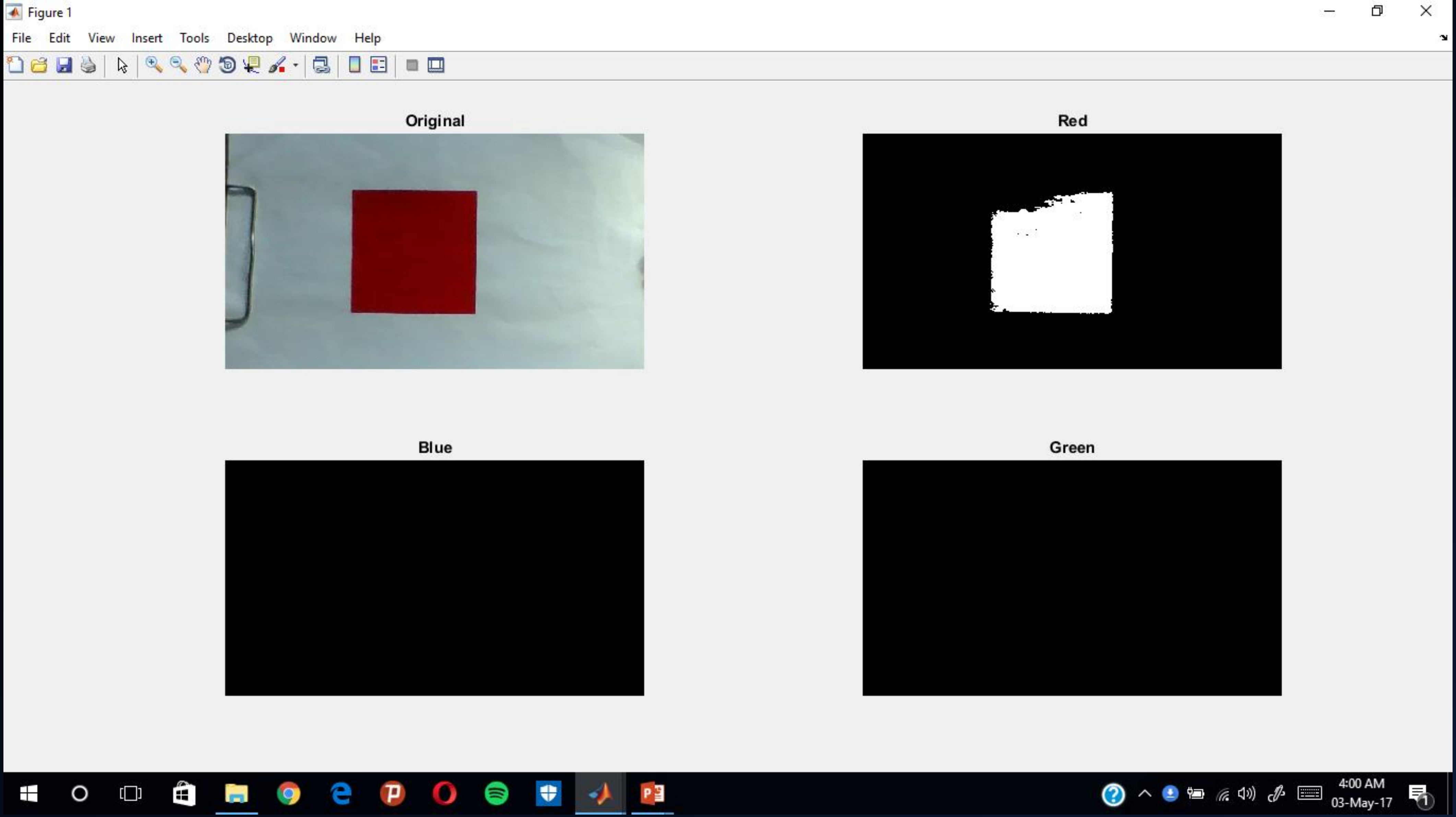


Matlab Code

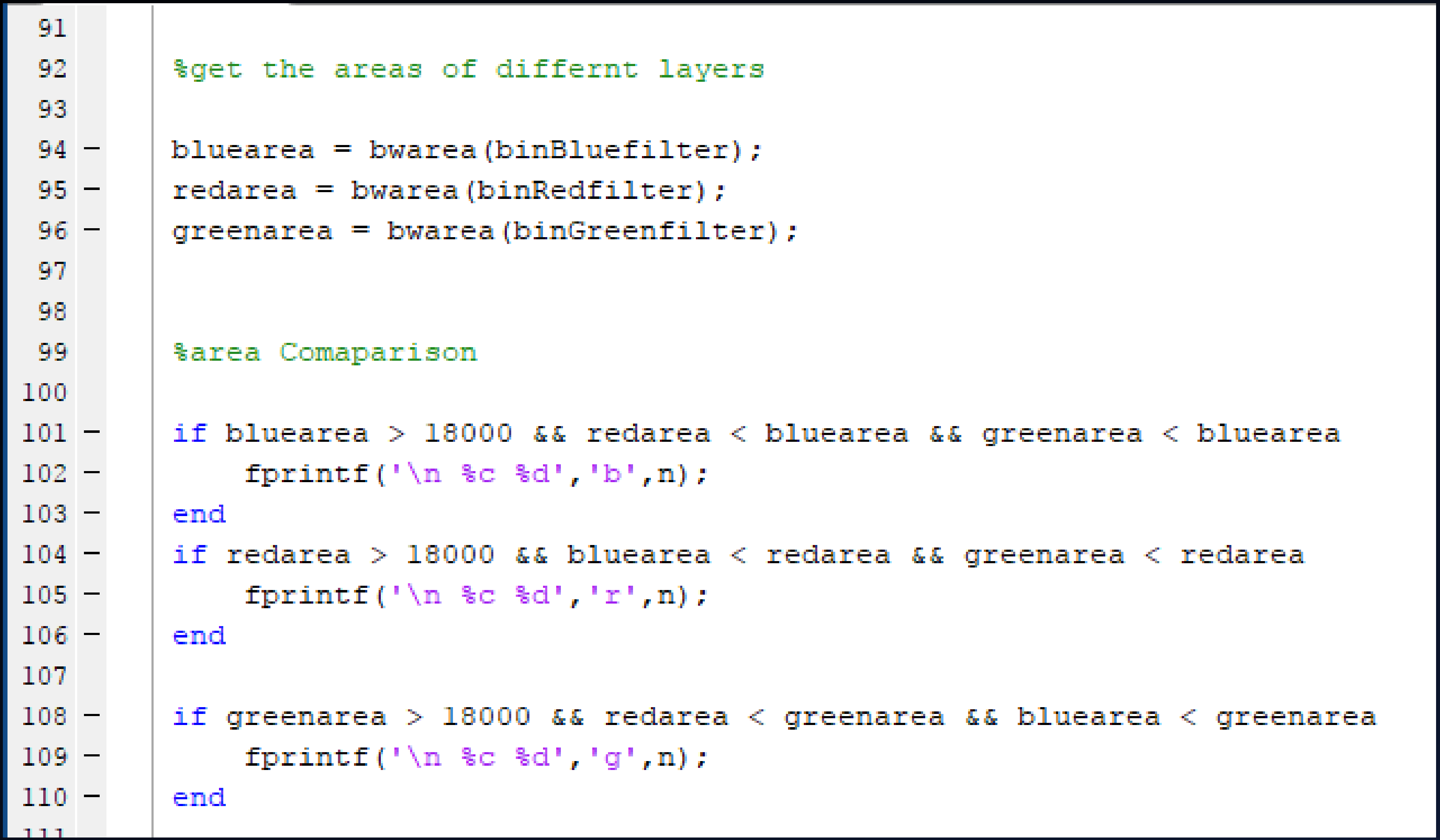


```
%binarize Red layer

binRed=zeros(r,c);
for i=1:r
    for j=1:c
        if R(i,j)>=100 && G(i,j)<=80&&B(i,j)<=80
            binRed(i,j)=255;
        end
    end
end

%filter noise

binRedfilter  = bwareaopen(binRed,6000);

%plot Red layer

subplot(2,2,2);
imshow(binRedfilter);
title('Red');
```

Binarized Image: Red

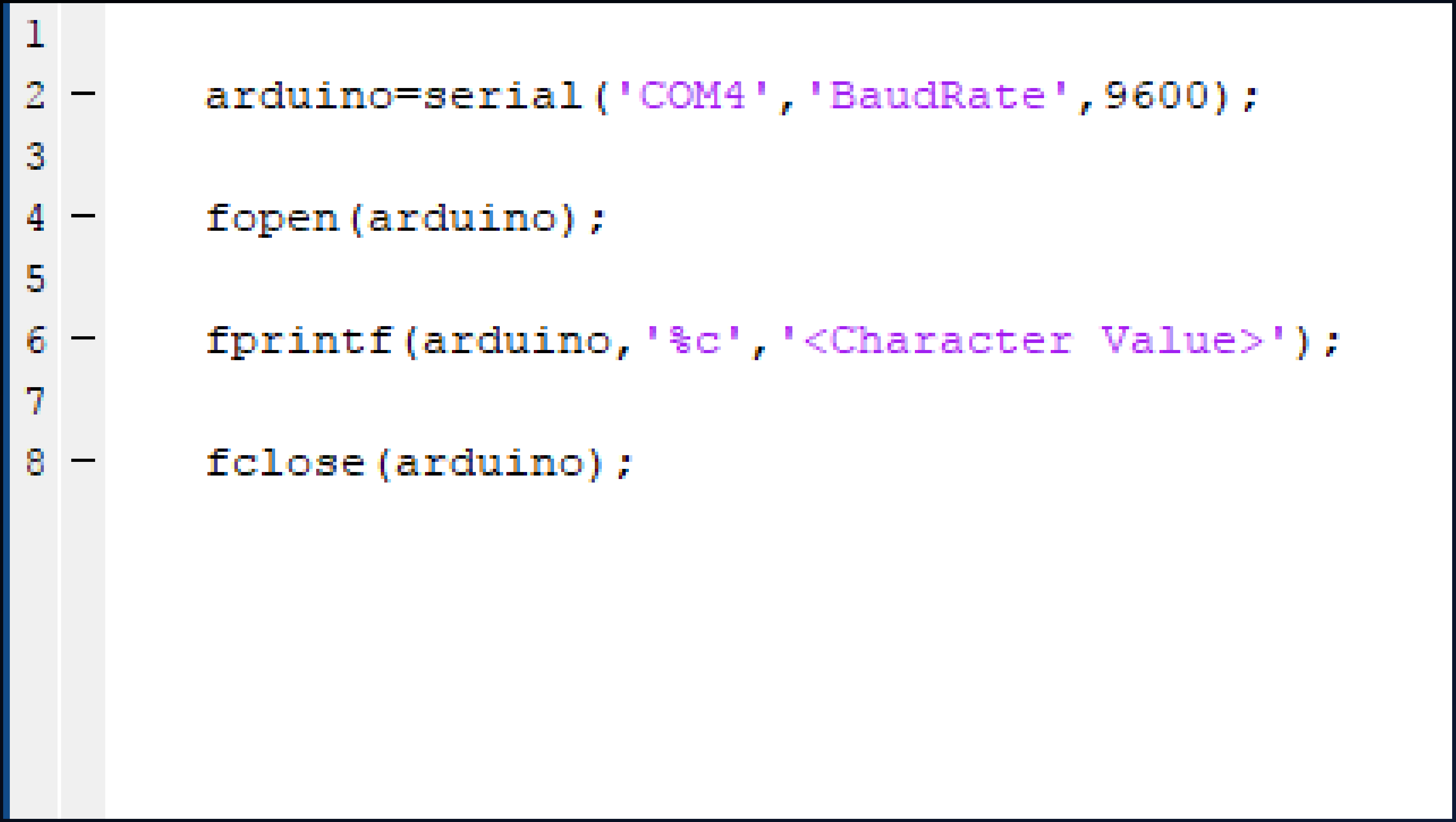


Matlab Code



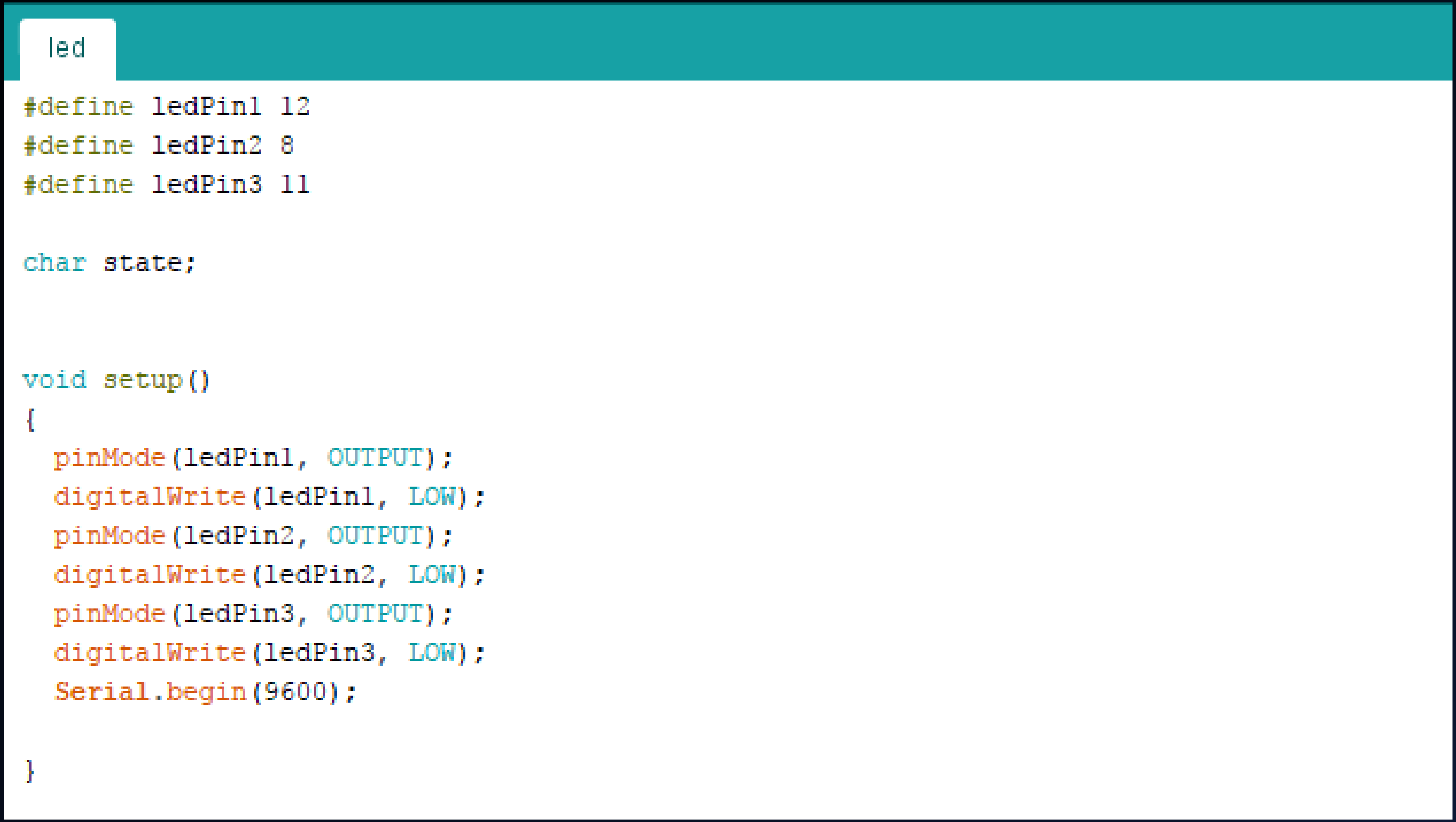
```

%get the areas of differnt layers

bluearea = bwarea(binBluefilter);
redarea = bwarea(binRedfilter);
greenarea = bwarea(binGreenfilter);


%area Comaparison

if bluearea > 18000 && redarea < bluearea && greenarea < bluearea
    fprintf('\n %c %d','b',n);
end
if redarea > 18000 && bluearea < redarea && greenarea < redarea
    fprintf('\n %c %d','r',n);
end

if greenarea > 18000 && redarea < greenarea && bluearea < greenarea
    fprintf('\n %c %d','g',n);
end
```

Matlab Code



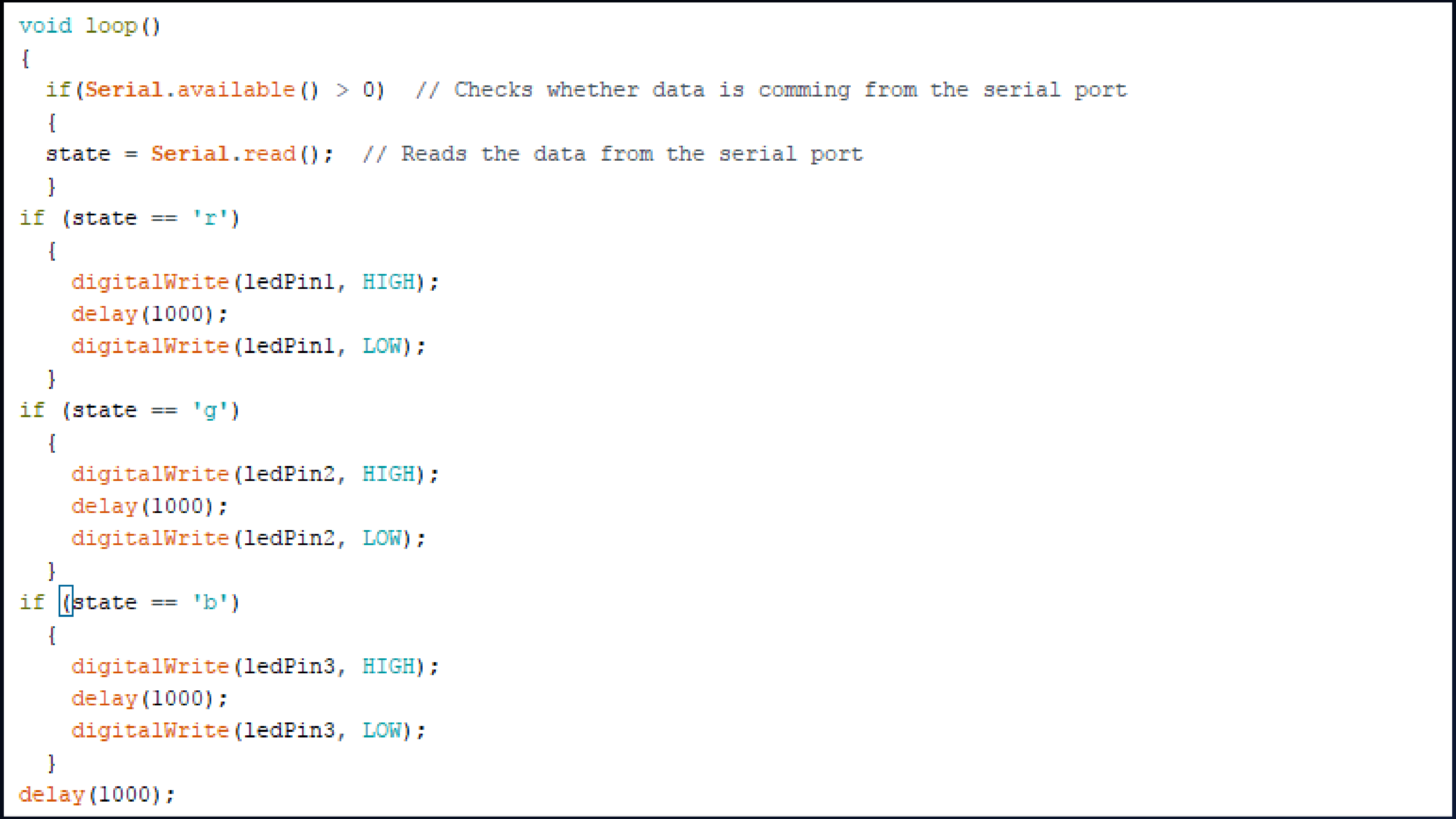
```

arduino=serial('COM4','BaudRate',9600);

fopen(arduino);

fprintf(arduino,'%c','<Character Value>');

fclose(arduino);
```

Arduino Code



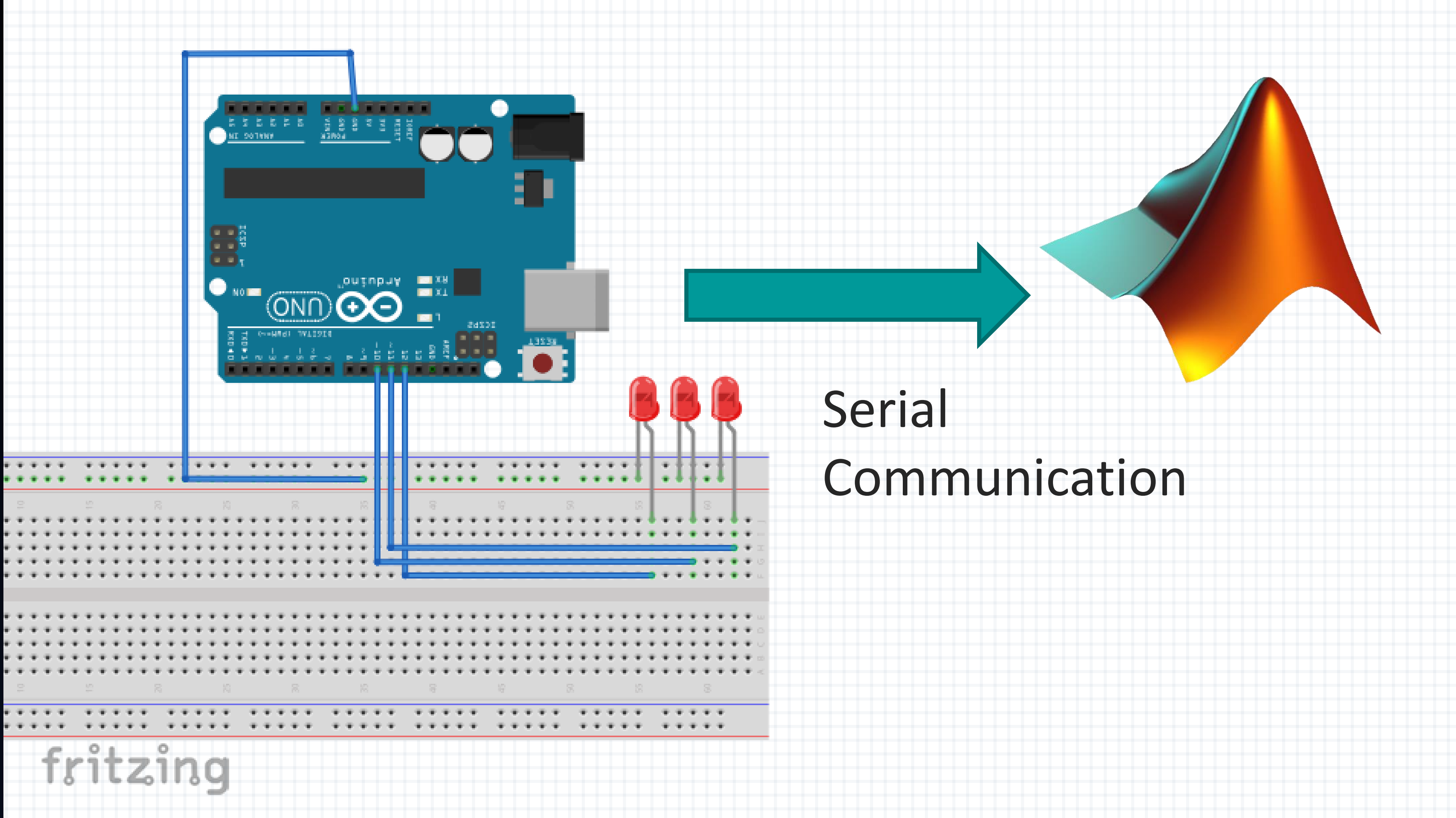
```
#define ledPin1 12
#define ledPin2 8
#define ledPin3 11

char state;


void setup()
{
  pinMode(ledPin1, OUTPUT);
  digitalWrite(ledPin1, LOW);
  pinMode(ledPin2, OUTPUT);
  digitalWrite(ledPin2, LOW);
  pinMode(ledPin3, OUTPUT);
  digitalWrite(ledPin3, LOW);
  Serial.begin(9600);

}
```

Arduino Code



```
void loop()
{
  if(Serial.available() > 0)  // Checks whether data is comming from the serial port
  {
  state = Serial.read();  // Reads the data from the serial port
  }
if (state == 'r')
  {
    digitalWrite(ledPin1, HIGH);
    delay(1000);
    digitalWrite(ledPin1, LOW);
  }
if (state == 'g')
  {
    digitalWrite(ledPin2, HIGH);
    delay(1000);
    digitalWrite(ledPin2, LOW);
  }
if (state == 'b')
  {
    digitalWrite(ledPin3, HIGH);
    delay(1000);
    digitalWrite(ledPin3, LOW);
  }
delay(1000);
```

Schematics



Serial Communication

Resources:

1. https://in.mathworks.com/help/matlab/
2. https://www.arduino.cc/en/Main/Documentation