Bilkent University

EE 391

MatLab Assignment 2

20/04/20

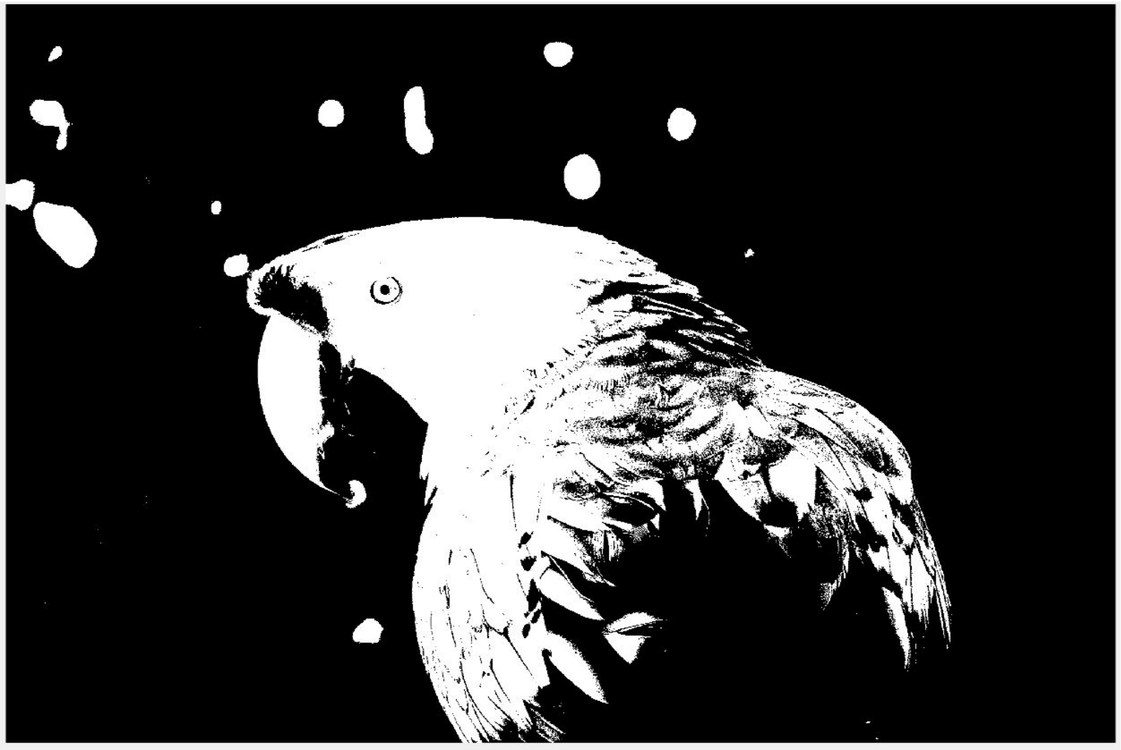
Okan Şen

21202377

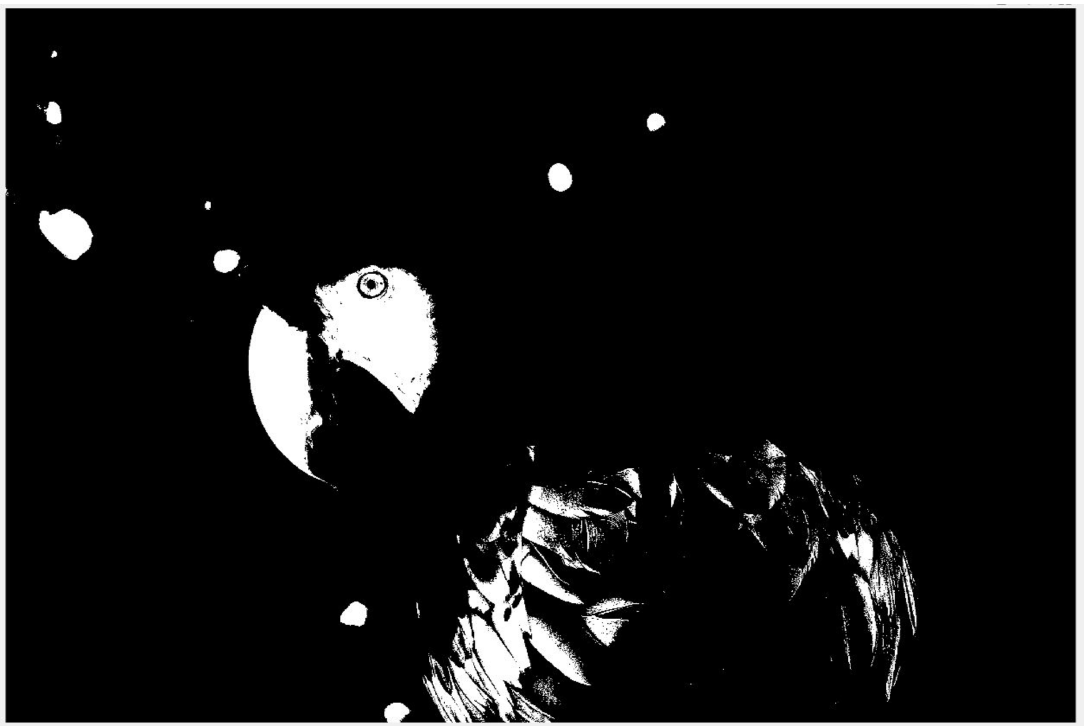
Section 2

Part 1

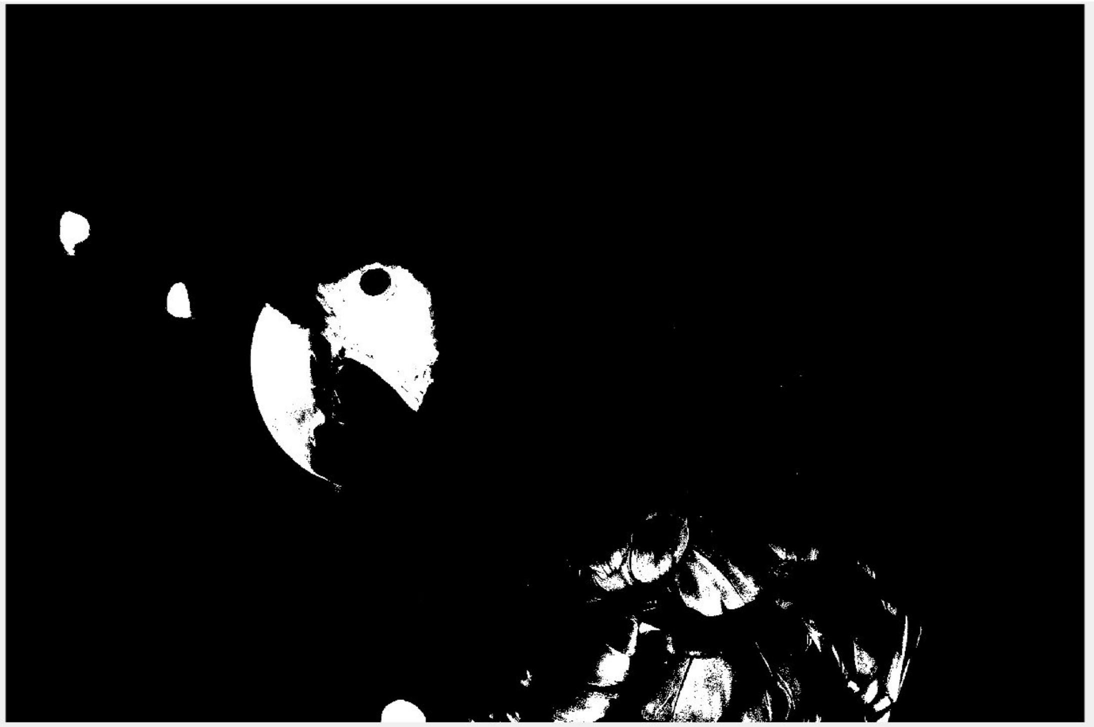
a) This loop takes the RGB values of each pixel and if the Red value of any pixel is greater than 140 in the Red Curve of the color palette, we assume the color should be 1 in the binary system. And vice versa as 0. When we print the binary colored version of the whole image we get this. SÎnce 140 is pretty in the middle areas of 255 numbered system, the result is distinguishable to the eye.



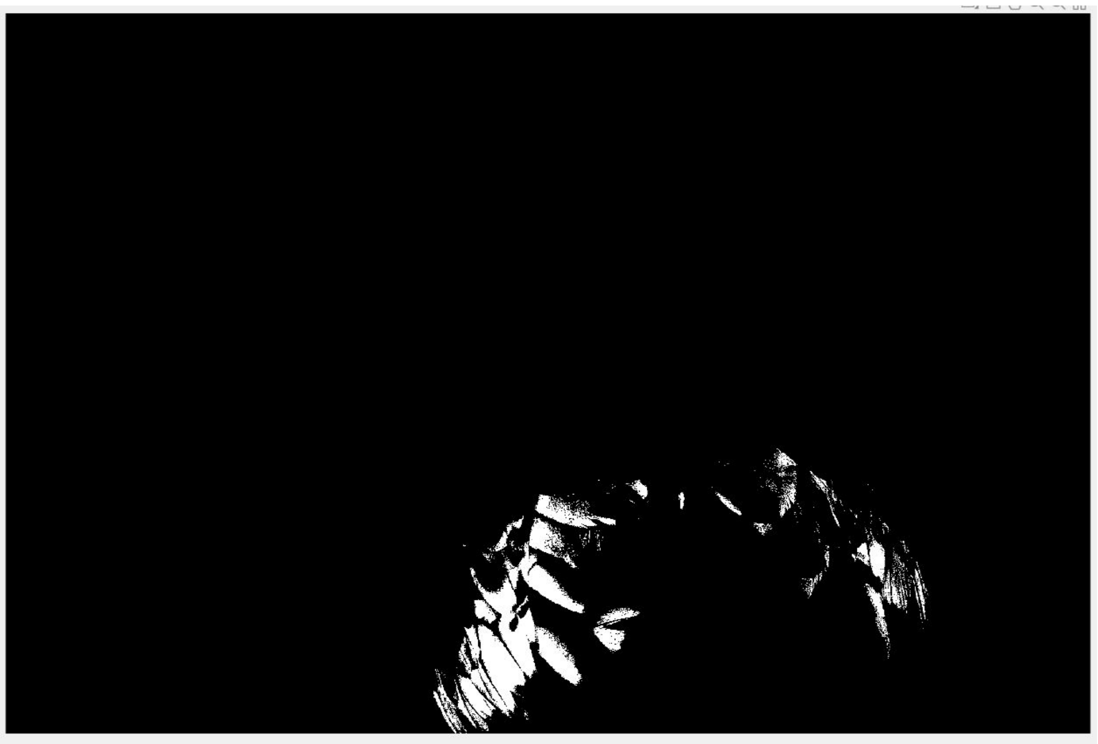
b) Here we do the same to the Green color. Apparently the greens in the picture are not strong enough to create a distinguishable image for the background, so they all are considered Zeroes. The Greens in the brighter areas still make for an understandable picture.



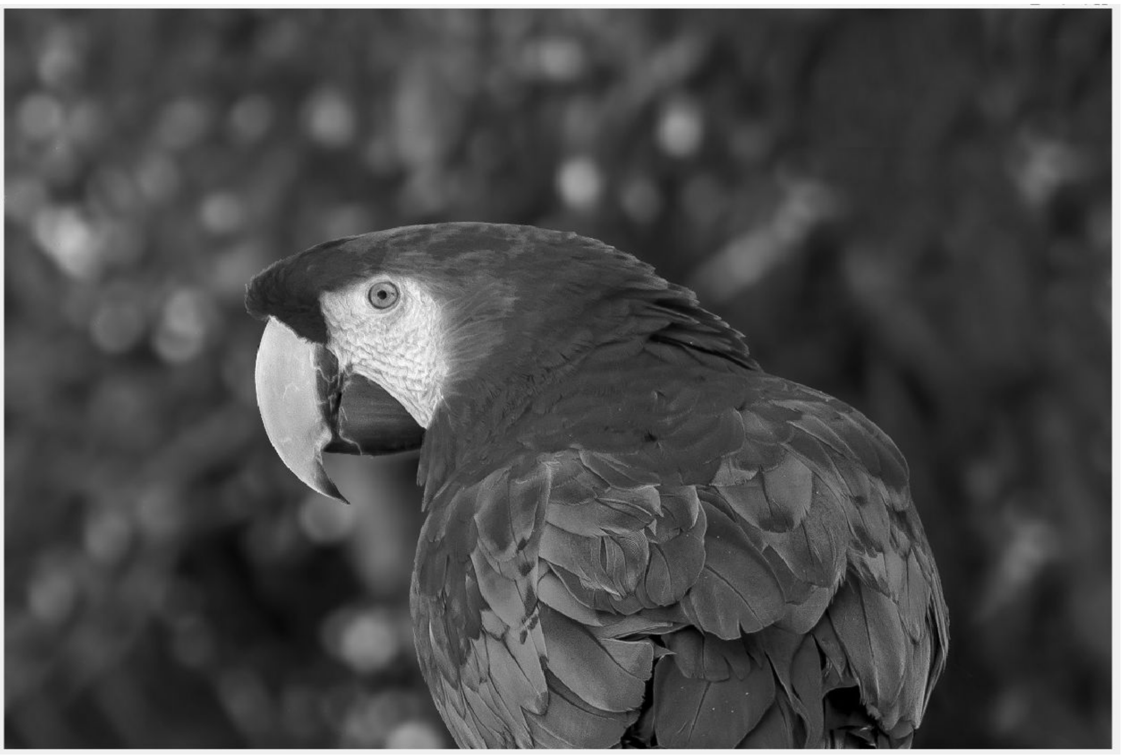
c) We do the same for blue, and for the same reasons, we get an even less distinguishable image.



d) For such specific values; $R > 140$, $B > 140$ and $G < 30$, we get only a handful of areas to be visible. The other areas are left in the pitch black.



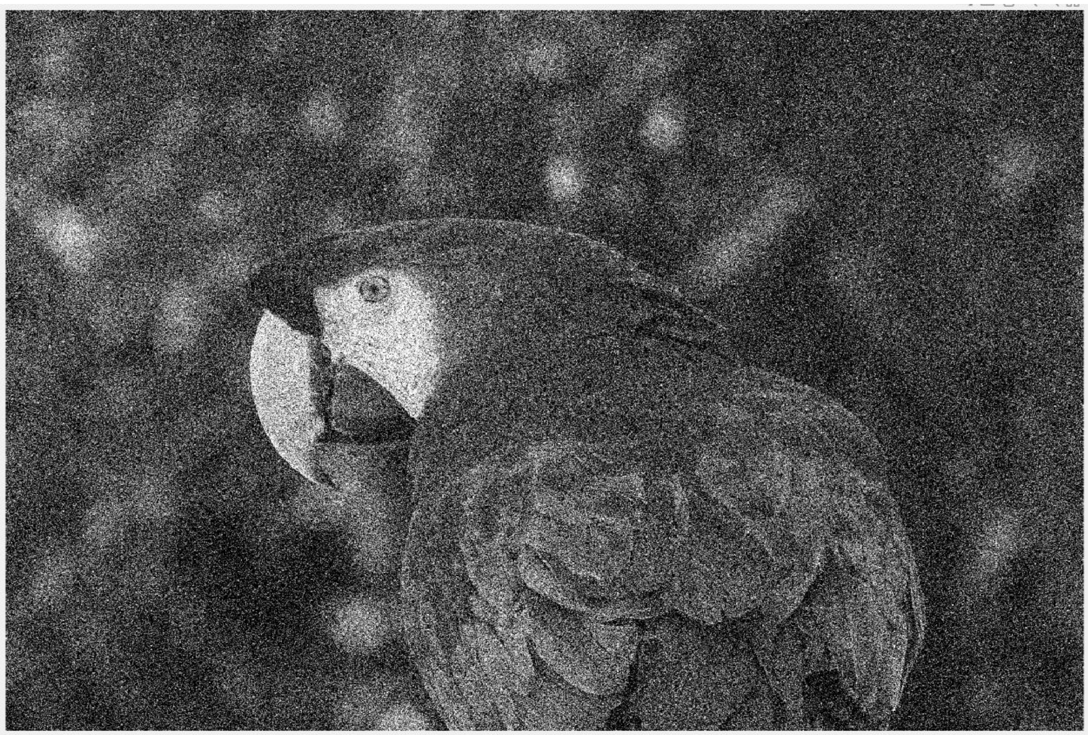
e) This is the greyscale image.



Part 2

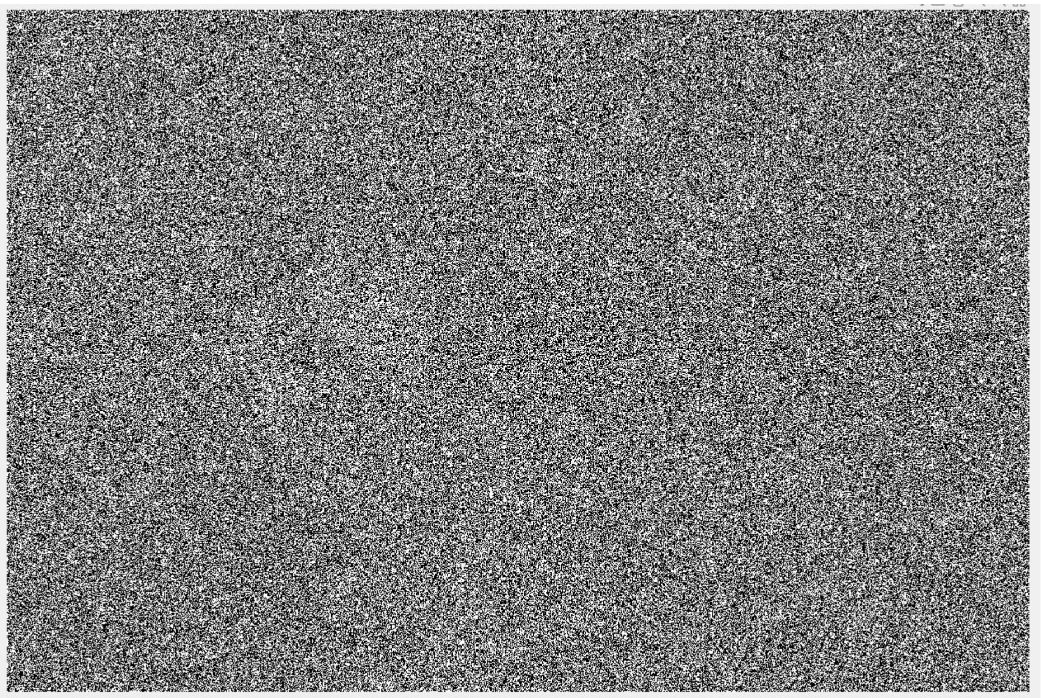
a) .

b) .



c) .

d) .



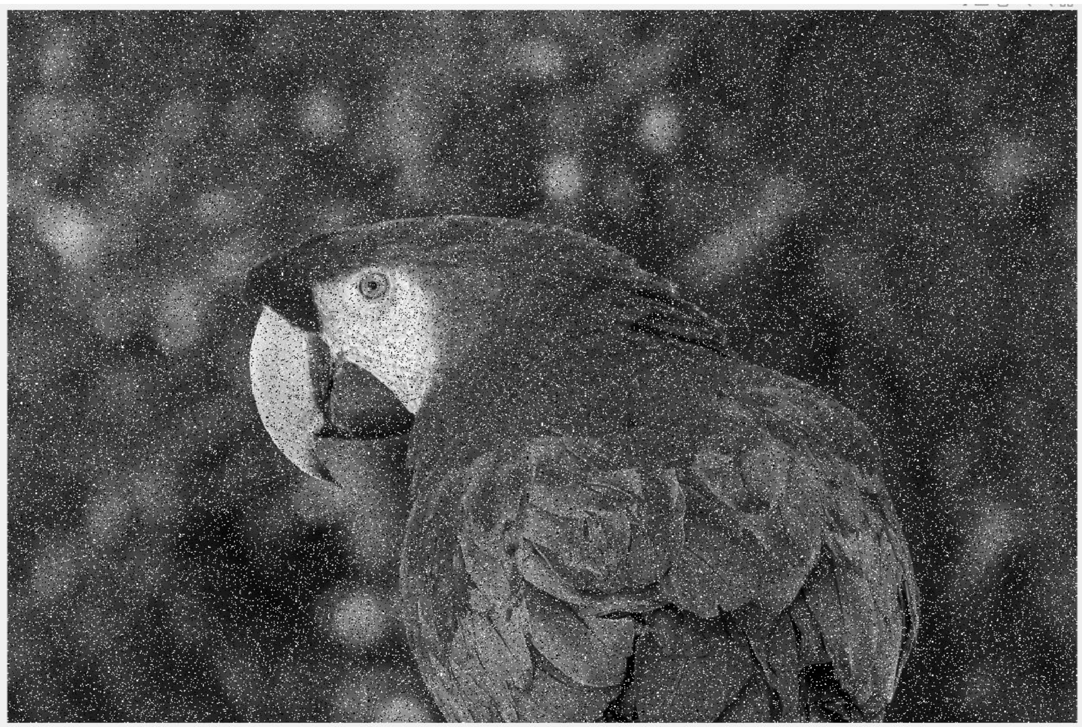
e) “noisypixels = rand(size(imgray,1), size(imgray,2)); “

This line creates a randomized noise profile for us, using the grey version of our original picture.

“imsaltnoise(find(noisypixels <= (1/16))) = 255;”
Then finds pixels that satisfy the conditions in the given line, and gives them a value of 255 in imsaltnoise image.

“imsaltnoise(find(noisypixels >= (15/16))) = 0;”
And does vice versa in this line.

This creates many random black and white dots on the image. Which we call salt and pepper noise.



f) .

Part 3

Part 4

Code

```
%% Program runs by itself with each part waiting for the previous one
%% for z seconds.
z = 1;
A = imread('image3.jpg');
A = double(A);
```

```
%%image(A);
[m,n,o] = size(A);

%% --------------------------------- PART I ---------------------------------
%%  ======== STEP A
%% Traverse through given image and put binary values
%% according to Red values being greater than 140 or not
for i = 1:m
   for j = 1:n
      for k = 1:3
         if A(i,j,1)>140
            Y(i,j,k) = 1;
         else
            Y(i,j,k) = 0;
         end
      end
   end
end
imshow(Y);
pause(z);
%%  ======== STEP B
%% according to Green values being greater than 140 or not
for i = 1:m
   for j = 1:n
      for k = 1:3
         if A(i,j,2)>140
            Y(i,j,k) = 1;
         else
            Y(i,j,k) = 0;
         end
      end
   end
end
imshow(Y);
pause(z);
%%  ======== STEP C
%% according to Blue values being greater than 140 or not
for i = 1:m
```

```
    for j = 1:n
        for k = 1:3
            if A(i,j,3)>140
                Y(i,j,k) = 1;
            else
                Y(i,j,k) = 0;
            end
        end
    end
end
imshow(Y);
pause(z);
%%  ======== STEP D
%% according to R > 140 G >140 B < 30
for i = 1:m
    for j = 1:n
        for k = 1:3
            if A(i,j,1)>140 && A(i,j,2)>140 && A(i,j,3)<30
                Y(i,j,k) = 1;
            else
                Y(i,j,k) = 0;
            end
        end
    end
end
imshow(Y);
pause(z);
%%  ======== STEP E
%% according to (R+G+B)/3
%gray(i,j);
for i = 1:m
    for j = 1:n

            gray(i,j) = ((A(i,j,1) + A(i,j,2) + A(i,j,3))/3);
            %%gray(i,j) = round(gray(i,j)/255);

    end
end
imshow(gray, []);
```

```
pause(z);

%gray = (A(:,:,1)+A(:,:,2)+A(:,:,3))/3; % This lin also gives the same
%result without adding any loop.
%imshow(gray, []);
%% --------------------------------- PART II --------------------------------
%%  ======== STEP A
I = gray;

%pause(z);
%%  ======== STEP B
imgray = gray;

gaussnoise = 64*randn(size(imgray,1), size(imgray,2));
imgaussnoise = uint8(double(imgray) + gaussnoise);

imshow(imgaussnoise, []);
pause(z);
%%  ======== STEP C


%%  ======== STEP D
imgray = gray;

gaussnoise = 1024*randn(size(imgray,1), size(imgray,2));
imgaussnoise = uint8(double(imgray) + gaussnoise);

imshow(imgaussnoise, []);
pause(z);
%%  ======== STEP E

imsaltnoise = gray;
noisypixels = rand(size(imgray,1), size(imgray,2));
imsaltnoise(find( noisypixels <= (1/16))) = 255;
imsaltnoise(find(noisypixels >= (15/16))) = 0;

imshow(imsaltnoise, []);
```