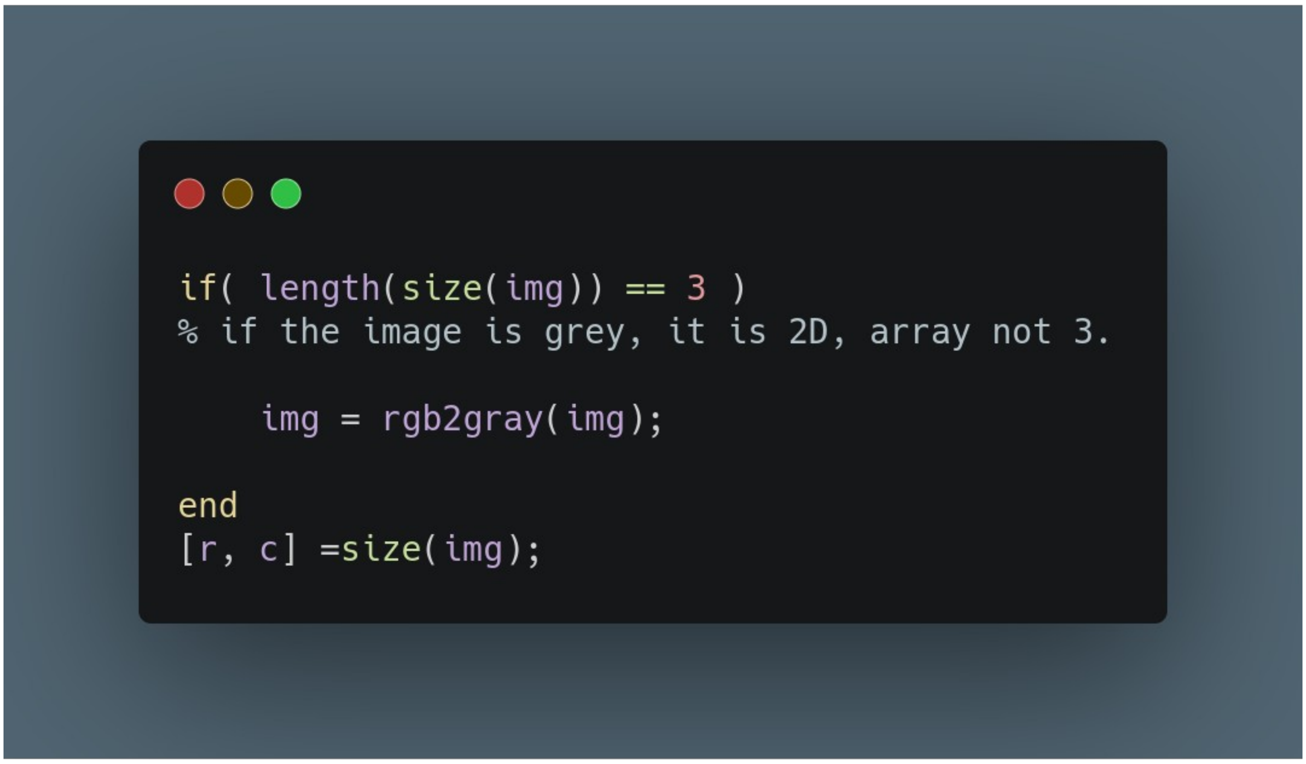
EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

Assingment 1:

Before starting, the following code is written because of any assumption cannot be made about image's type. However, one of my mistakes is, instead of creating another MATLAB script to check validity of image and converting them, I just add this part to begging of some of my scripts.

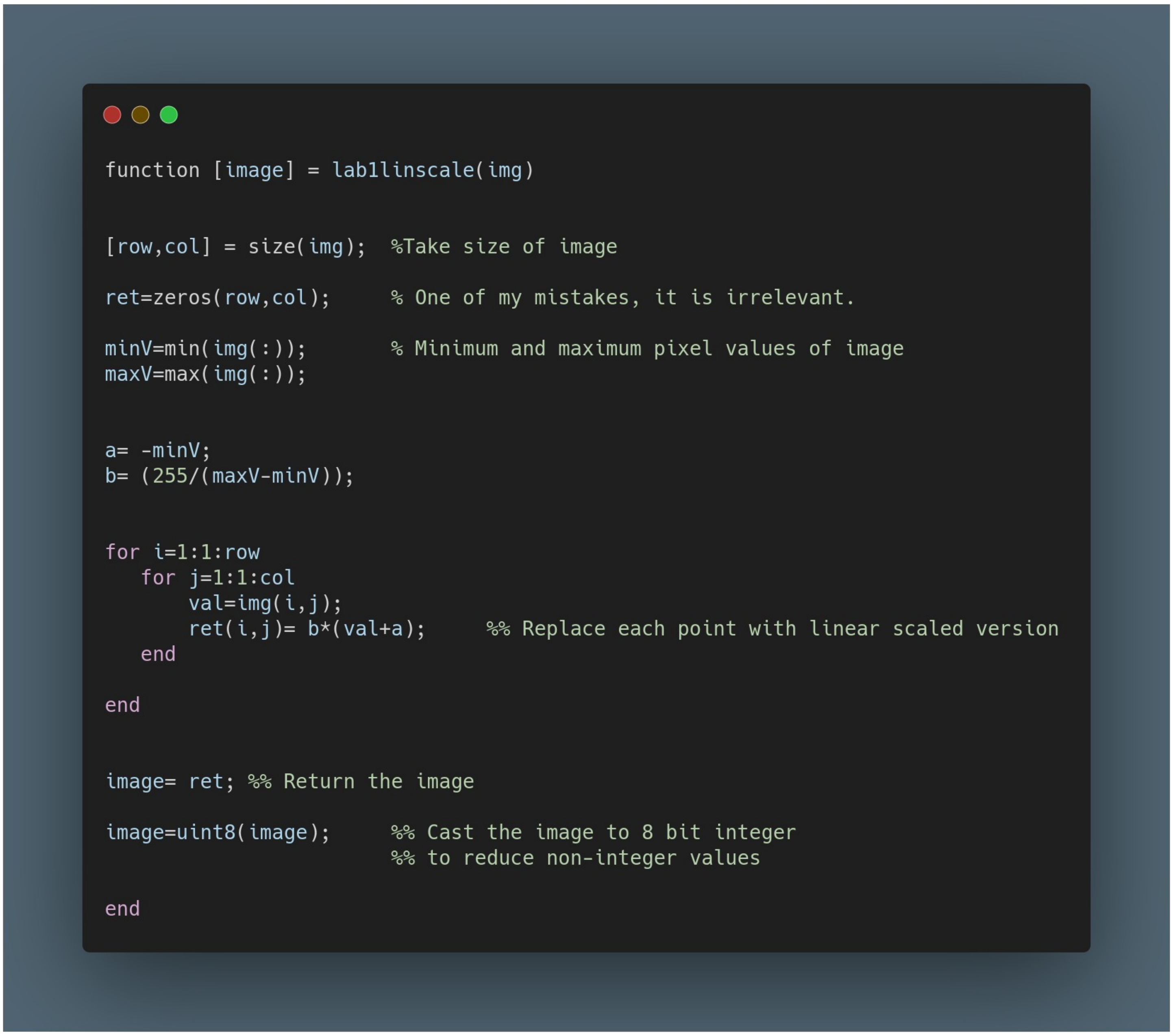


```
if( length(size(img)) == 3 )
% if the image is grey, it is 2D, array not 3.

    img = rgb2gray(img);

end
[r, c] =size(img);
```

In first task, what we are trying to achieve is scaling the image linearly. Linear scaling is a point operation which arrange histogram of image and make him stretched across the possible pixel values, - 8 bit pixels == 0-255 value range.

To linear scale we use the following formula:
g(u)= b*(u-a) where a is minimum value and b = Gmax / (u_max-u_min)
Gmax= 255 because maximum value a pixel can ever take is 255. In the next page, you can find my all code.

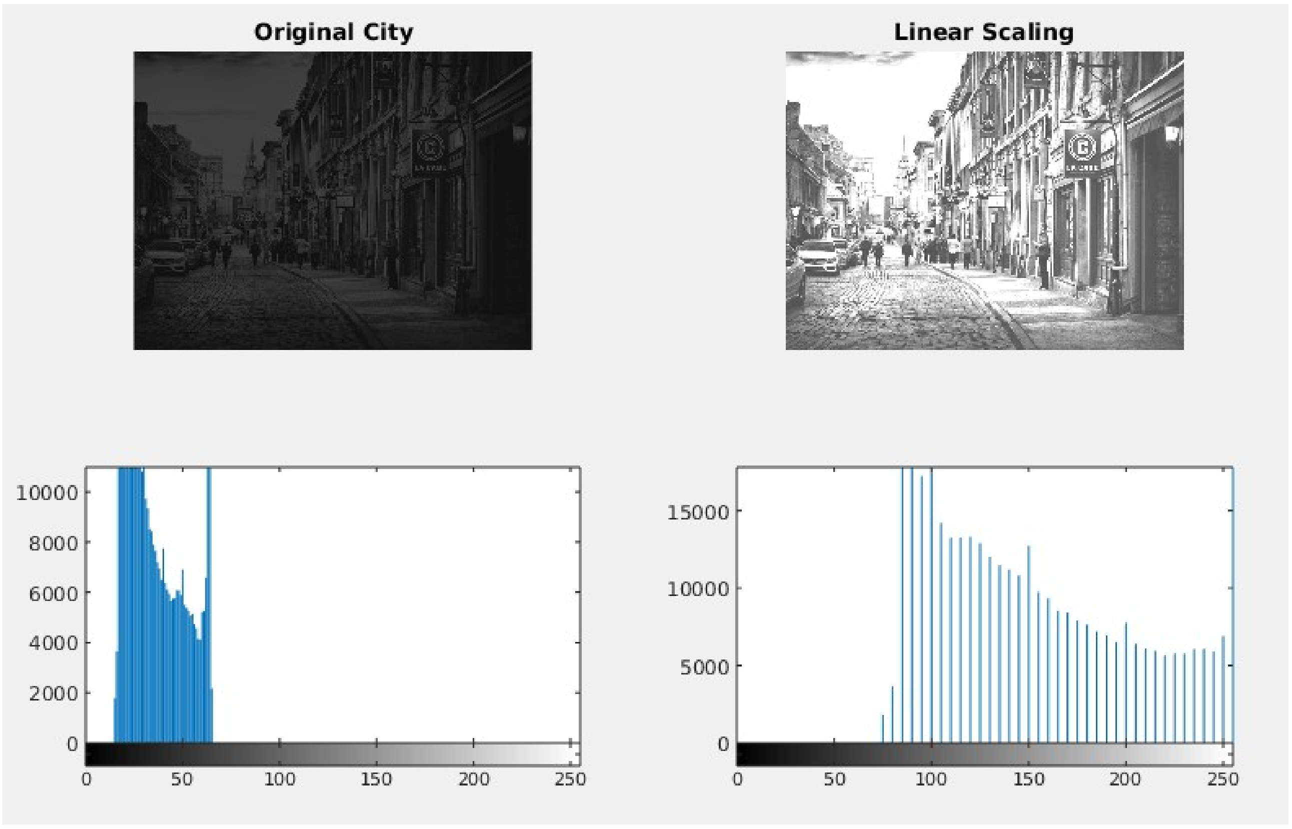
EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19



```
function [image] = lab1linscale(img)


[row,col] = size(img);  %Take size of image

ret=zeros(row,col);     % One of my mistakes, it is irrelevant.

minV=min(img(:));       % Minimum and maximum pixel values of image
maxV=max(img(:));


a= -minV;
b= (255/(maxV-minV));


for i=1:1:row
   for j=1:1:col
       val=img(i,j);
       ret(i,j)= b*(val+a);     %% Replace each point with linear scaled version
   end

end


image= ret; %% Return the image

image=uint8(image);     %% Cast the image to 8 bit integer
                        %% to reduce non-integer values

end
```

In the next page you can find result. However, It does not seems like linearly scaled. Unfortunately, I have failed the first task. Please check the results.

EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

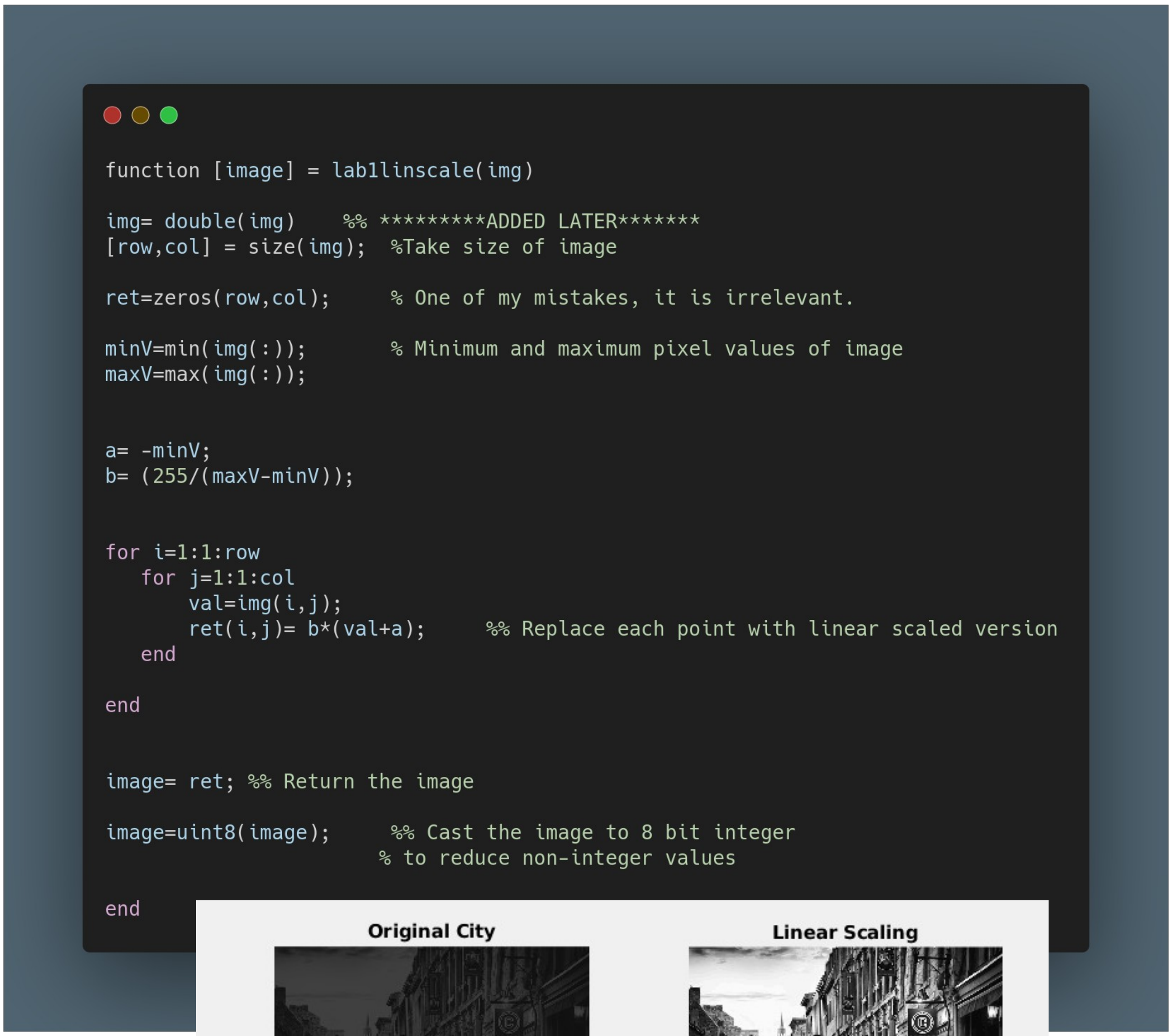


The mistake that I made:

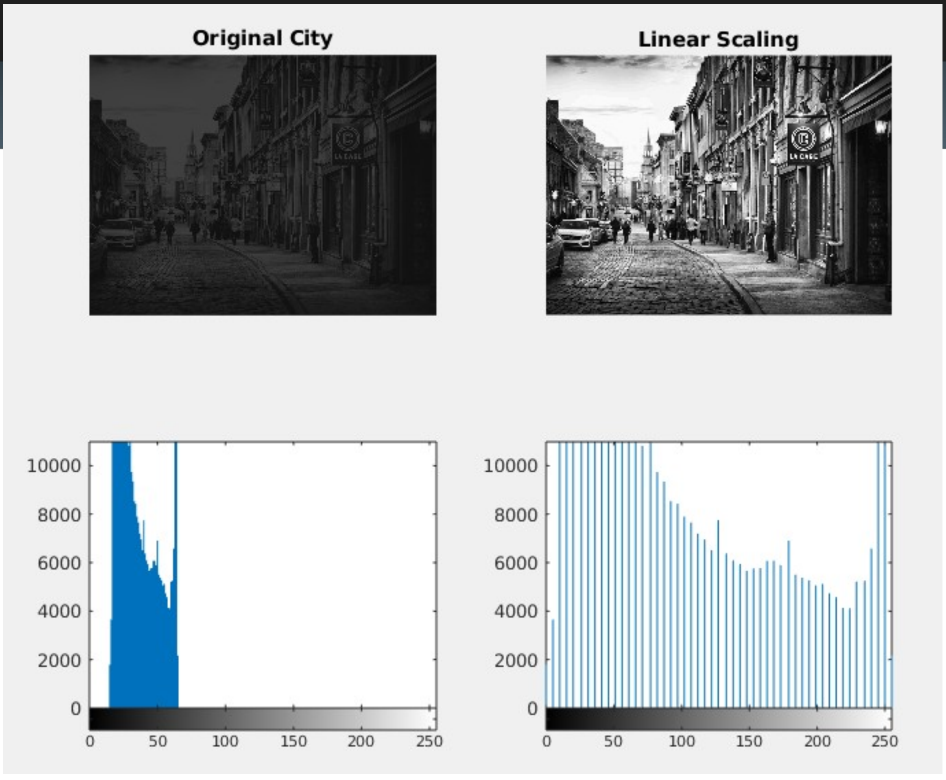
In my submitted code, I have to forgot to convert image to double values. Result of that, I calculated incorrect ‘b’ value and failed the linear scaling.

However, I corrected the code and obtain proper results. Only one line added corrected version of my code and results placed in below.

EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19



```
function [image] = lab1linscale(img)

img= double(img)    %% *********ADDED LATER*******
[row,col] = size(img);  %Take size of image

ret=zeros(row,col);     % One of my mistakes, it is irrelevant.

minV=min(img(:));       % Minimum and maximum pixel values of image
maxV=max(img(:));


a= -minV;
b= (255/(maxV-minV));


for i=1:1:row
   for j=1:1:col
       val=img(i,j);
       ret(i,j)= b*(val+a);      %% Replace each point with linear scaled version
   end

end


image= ret; %% Return the image

image=uint8(image);      %% Cast the image to 8 bit integer
                        % to reduce non-integer values

end
```



EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

Self-Critisism for Task 1:

- I could not manage to time well and forgot to put histograms of images to my lab submission.
- I did not put any comment on my codes due to time limitation and stress of my first lab.
- I created an 2D image matrix by zeros to use later, however, forgot the delete and it just seems irrelevant.
- There was a division operation so I could get non-integer results. I missed that and instead of casting the pixels to double, I used the integer matrix and get wrong result. Also, lack of histograms let me believe that I manage the task 1 successfully.
- I unintentionally assumed the input image will be gray-scale by looking at the example image. Therefore, my code would not work with RGB images.
- Nevertheless, I implement the formula and get a result successfully.

Assignment 2:

Conditional scaling is another point operation which scale the histogram of image just like linear scaling. In additional, conditional scaling equalize the variance and mean to other reference image. After the scaling, reference image and the image that worked on, are going to have same variance and mean values. The formula to achieve that is

a = meanJ * (stdI / stdJ)- meanI and b= stdJ / stdI => g(u) = b(u+a)

In spite of assignment 1, this model worked successfully. No correction is required. Below you can find working model, and source code.

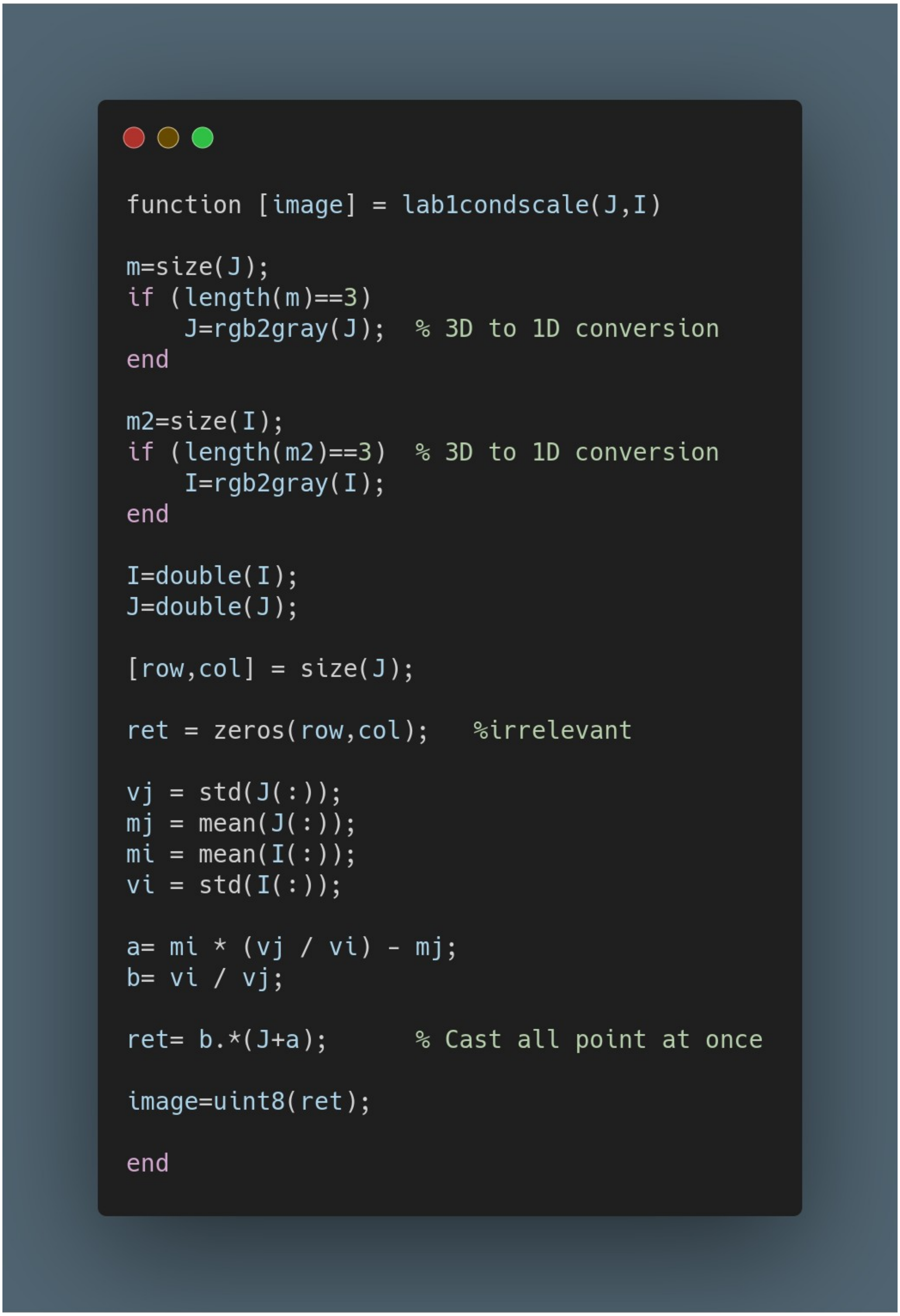
Self-critisim:

- This time, I handled RGB images well. Any kind of image will be valid.
- I initialize the image that will return, by “=zeros()” , it was not necessary
- Nevertheless, there isn’t any comment.
- There is no mean and variance value in figure.

EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

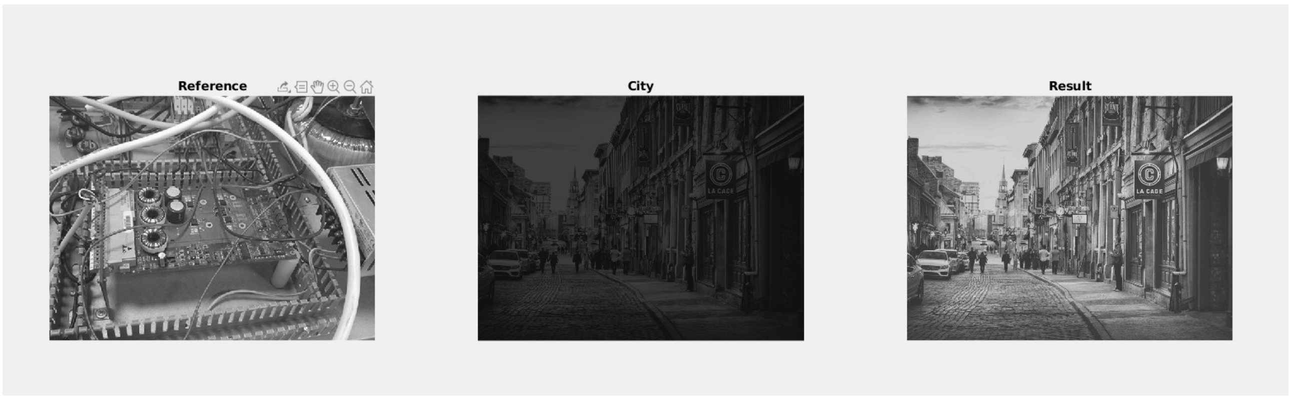
EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19



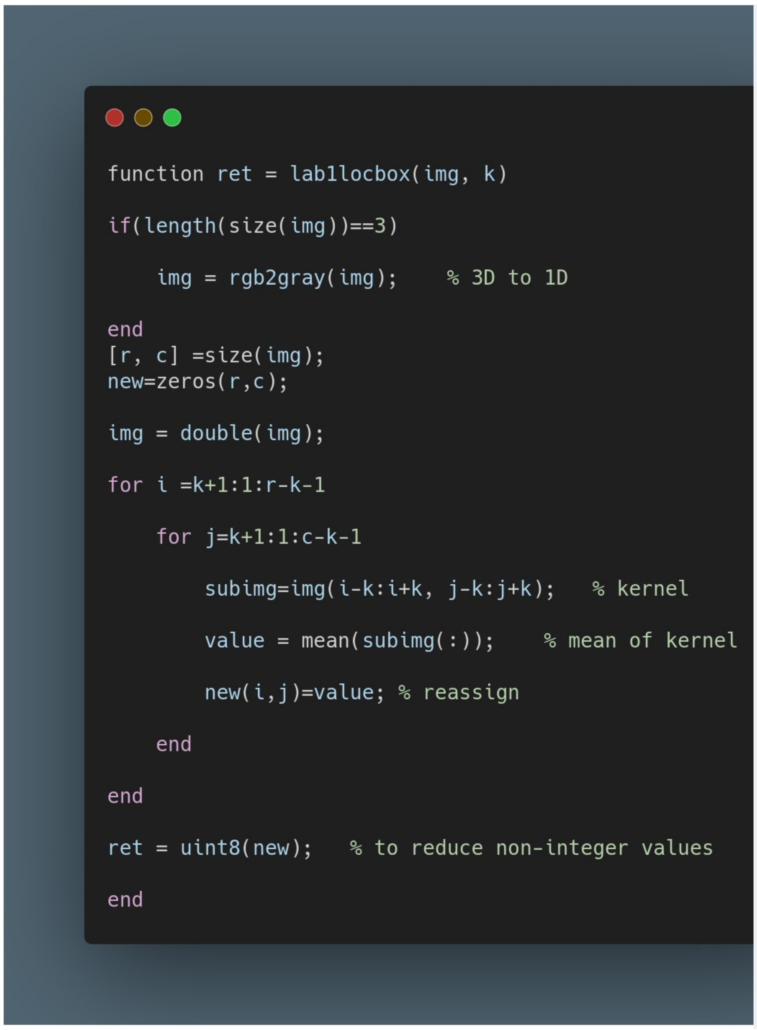
```
function [image] = lab1condscale(J,I)

m=size(J);
if (length(m)==3)
    J=rgb2gray(J);  % 3D to 1D conversion
end

m2=size(I);
if (length(m2)==3)  % 3D to 1D conversion
    I=rgb2gray(I);
end

I=double(I);
J=double(J);

[row,col] = size(J);

ret = zeros(row,col);   %irrelevant

vj = std(J(:));
mj = mean(J(:));
mi = mean(I(:));
vi = std(I(:));

a= mi * (vj / vi) - mj;
b= vi / vj;

ret= b.*(J+a);      % Cast all point at once

image=uint8(ret);

end
```

EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

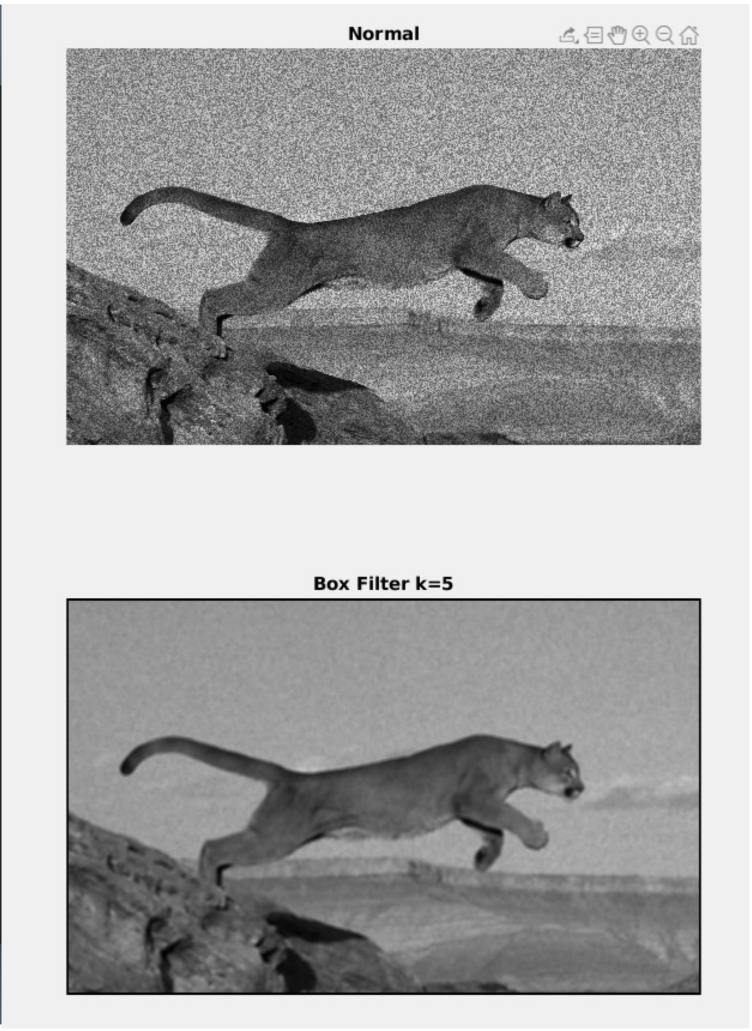


Assignment 3:

Local mean filter is a local operator which takes a mean of a kernel and reassigns the result to proper point. It was crucial for reducing the noises, however, proper kernel size is required.



```
function ret = lab1locbox(img, k)

if(length(size(img))==3)

    img = rgb2gray(img);    % 3D to 1D

end
[r, c] =size(img);
new=zeros(r,c);

img = double(img);

for i =k+1:1:r-k-1

    for j=k+1:1:c-k-1

        subimg=img(i-k:i+k, j-k:j+k);   % kernel

        value = mean(subimg(:));    % mean of kernel

        new(i,j)=value; % reassign

    end

end

ret = uint8(new);   % to reduce non-integer values

end
```



EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

Self-Criticism:

- No comment either… (due to lack of time management and being unfamiliar to labs)
- I am proud of myself, all other things seems great to me.

Assignment 4

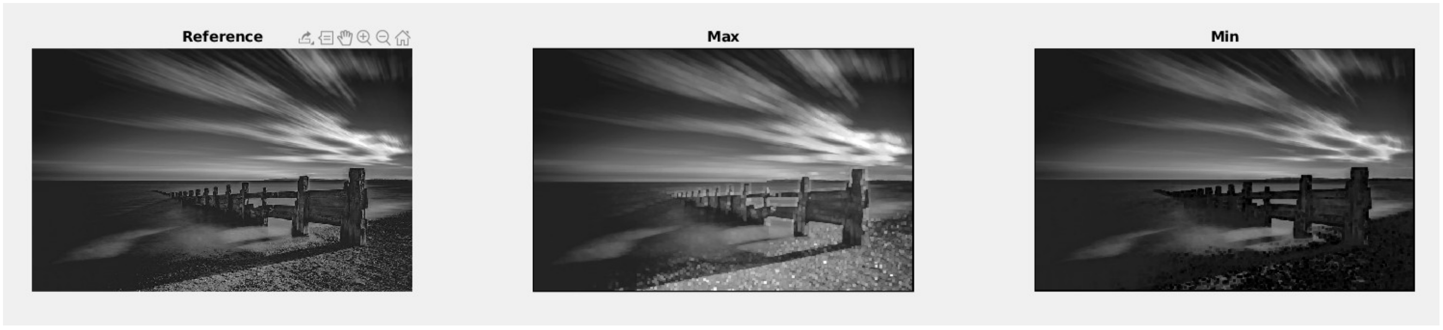
Local max and min filter is a local operator which takes a max/min of a kernel and reassigns the result to proper point. It is too similar to assignment 3. In fact, I just copied and pasted my previous code and changed one line to make assignment 4. The usage of this filter getting darker or brighter of the image as well as bluring the image. Also it may reduce the noises as well but not effectively.

I added also third variable near of image and kernel size value. It is a one bit variable which means ‘1’ for minimum filter and ‘2’ for maximum filter.

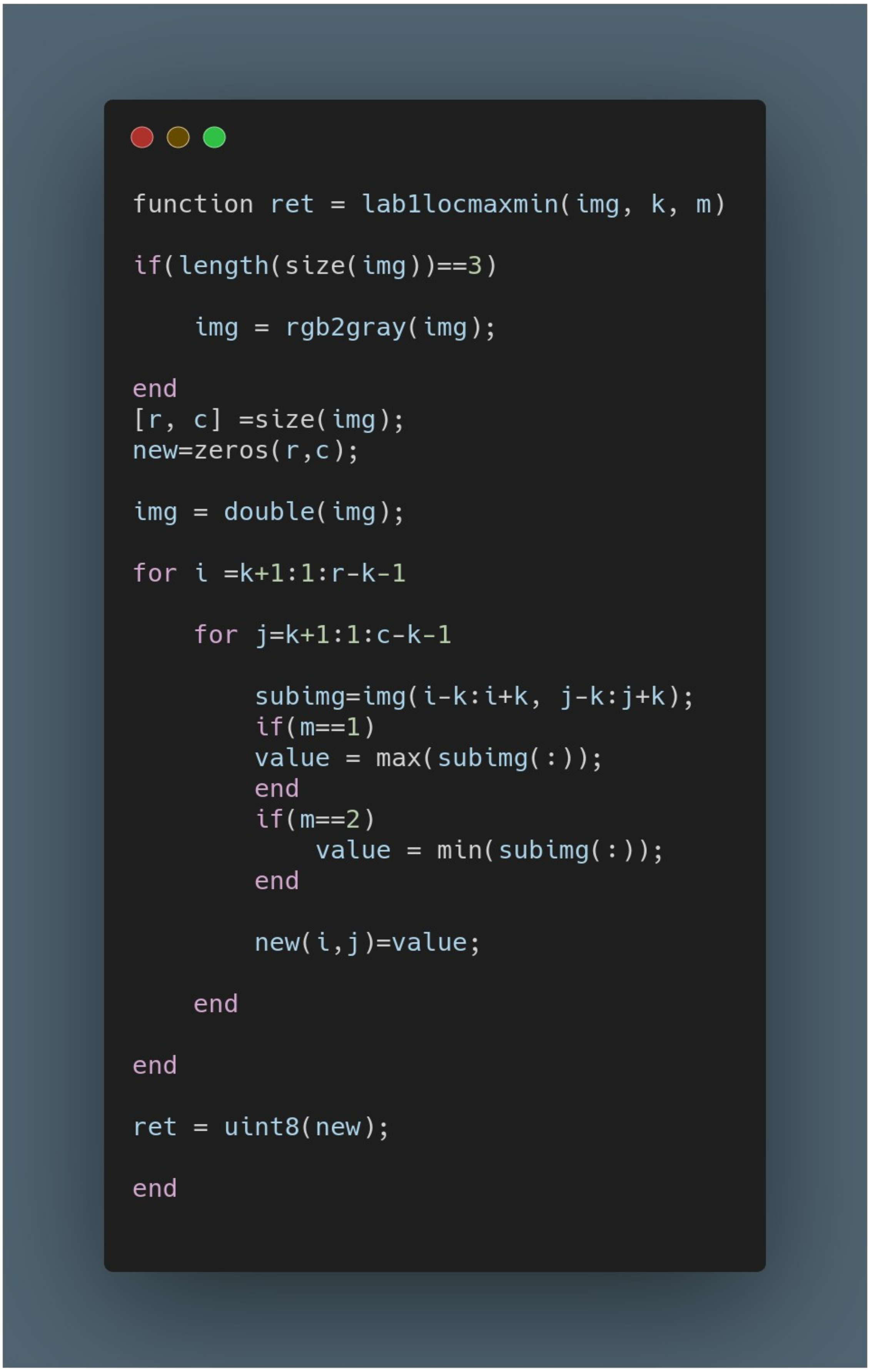
Self-Criticism:

- No comment to explain what code does.
- User-defined third variable may be bad idea due to readability.
- There was no other mistake that I spotted, it seems successful implementation of local max min filter.

You can find the figure and the source code below.



EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

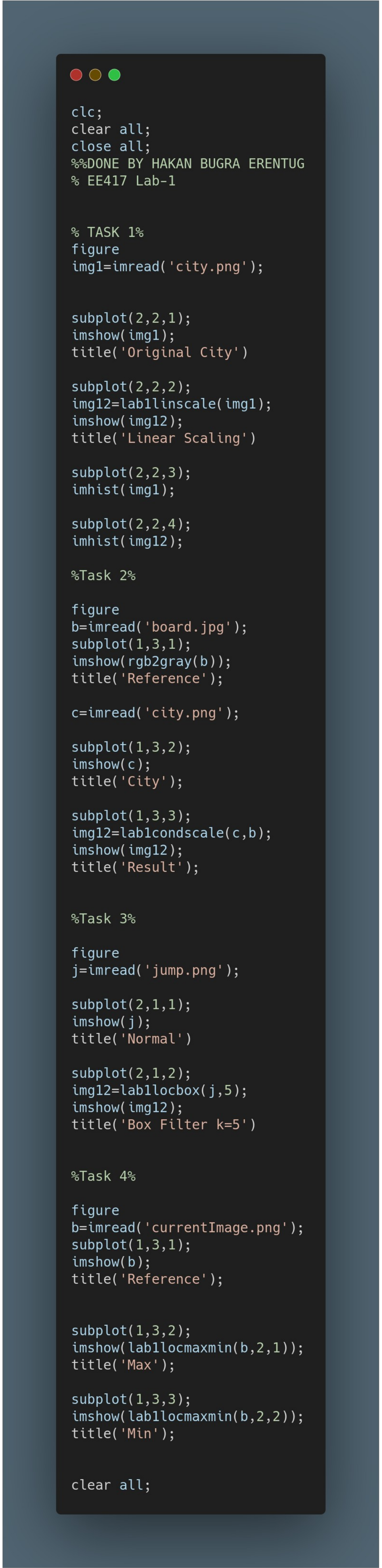


```
function ret = lab1locmaxmin(img, k, m)

if(length(size(img))==3)

    img = rgb2gray(img);

end
[r, c] =size(img);
new=zeros(r,c);

img = double(img);

for i =k+1:1:r-k-1

    for j=k+1:1:c-k-1

        subimg=img(i-k:i+k, j-k:j+k);
        if(m==1)
        value = max(subimg(:));
        end
        if(m==2)
            value = min(subimg(:));
        end

        new(i,j)=value;

    end

end

ret = uint8(new);

end
```

EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19

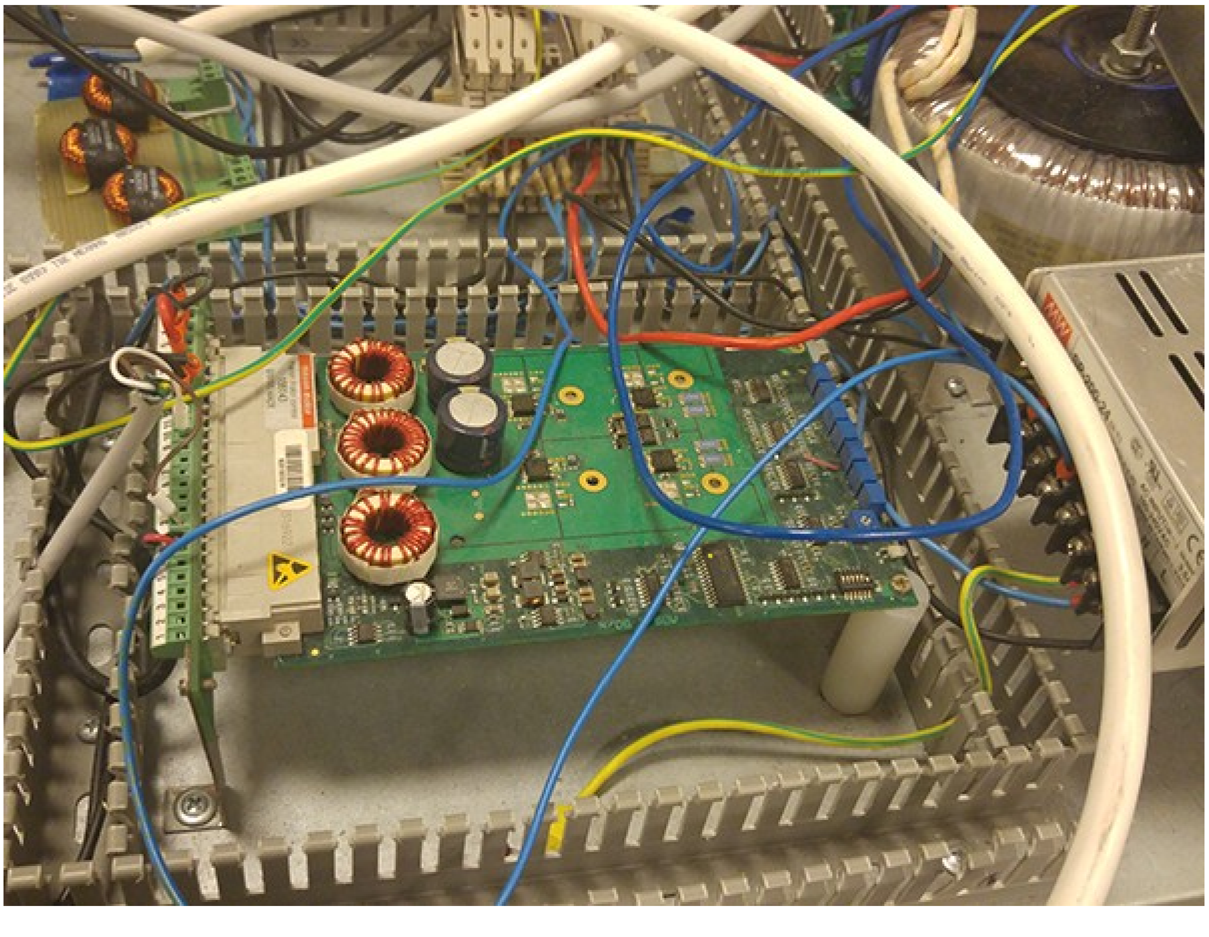
Appendices:

The only main code (The executer) has placed in appendix part. Because I already show the other codes that I used in project. Also, all unmodified images has been show below.



```
clc;
clear all;
close all;
%%DONE BY HAKAN BUGRA ERENTUG
% EE417 Lab-1


% TASK 1%
figure
img1=imread('city.png');


subplot(2,2,1);
imshow(img1);
title('Original City')

subplot(2,2,2);
img12=lab1linscale(img1);
imshow(img12);
title('Linear Scaling')

subplot(2,2,3);
imhist(img1);

subplot(2,2,4);
imhist(img12);

%Task 2%

figure
b=imread('board.jpg');
subplot(1,3,1);
imshow(rgb2gray(b));
title('Reference');

c=imread('city.png');

subplot(1,3,2);
imshow(c);
title('City');

subplot(1,3,3);
img12=lab1condscale(c,b);
imshow(img12);
title('Result');


%Task 3%

figure
j=imread('jump.png');

subplot(2,1,1);
imshow(j);
title('Normal')

subplot(2,1,2);
img12=lab1locbox(j,5);
imshow(img12);
title('Box Filter k=5')


%Task 4%

figure
b=imread('currentImage.png');
subplot(1,3,1);
imshow(b);
title('Reference');


subplot(1,3,2);
imshow(lab1locmaxmin(b,2,1));
title('Max');

subplot(1,3,3);
imshow(lab1locmaxmin(b,2,2));
title('Min');


clear all;
```

EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19



EE417- Computer Vision Lab 1 Report
Hakan Buğra Erentuğ 23637
08 / 10 / 19