Figure 1 Cankaya University Logo

EE-462 Introduction to Image Processing

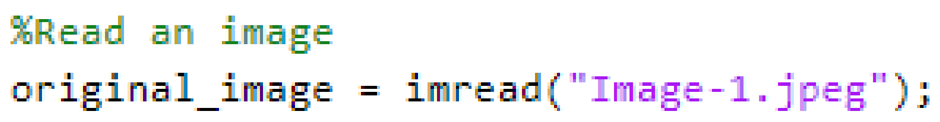
Faik Doruk Akgüney

201726003

Our project is writing a MATLAB program that can read an any sized image with predefined name from predefined location path, then creates a sketch of that image as output where background is white and sketchings are black. We start the project by first searching for the source related to the task requested from us. Then we create the algorithm of the program we will write. We start to write the program according to this algorithm. The algorithm I will use is as follows:

1. Write the image to the program.
2. Converting the image to grayscale format.
3. Find the negative of a grayscale image.
4. Blurring a negative grayscale image with a gaussian filter
5. Finding the negative of the blurred image.
6. Obtaining the desired image by dividing the grayscale image into a negative blurred image.

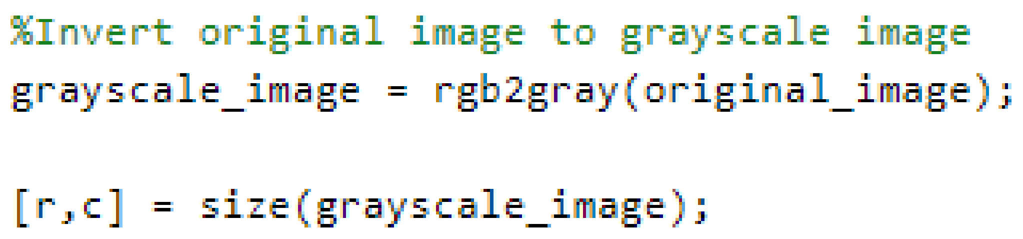
We start the program with the close all, clear all, clc commands. If there is a program that has been run before, we clean the figures, workspace, and command window. We transfer the image we use to MATLAB with the imread command.



```
%Read an image
original_image = imread("Image-1.jpeg");
```

Figure 2 Read an image

We convert the image we receive to grayscale format with the rgb2gray command. We find the row and column values of the image we converted to use when converting images to negative in the future.

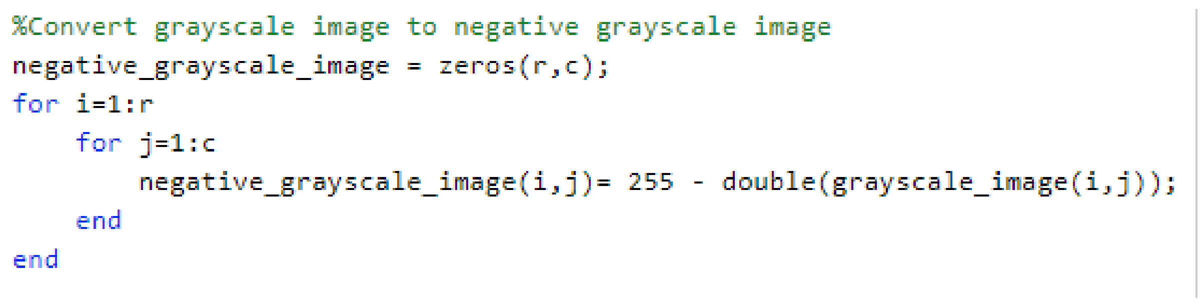


```
%Invert original image to grayscale image
grayscale_image = rgb2gray(original_image);

[r,c] = size(grayscale_image);
```

Figure 3 RGB to grayscale

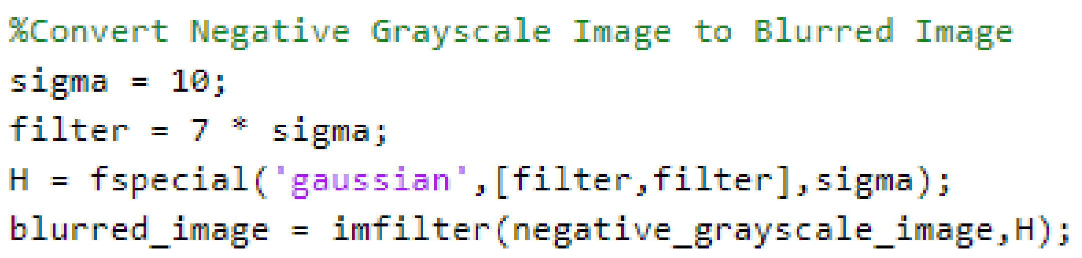
We find the negative of the grayscale image we found. We create a zero-matrix containing 0 as the row and column value of the image we just found. Then, with two for loops, we find the negative of each pixel one by one and throw them into the zero matrix we just created.



```
%Convert grayscale image to negative grayscale image
negative_grayscale_image = zeros(r,c);
for i=1:r
    for j=1:c
        negative_grayscale_image(i,j)= 255 - double(grayscale_image(i,j));
    end
end
```

Figure 4 Grayscale to negative grayscale

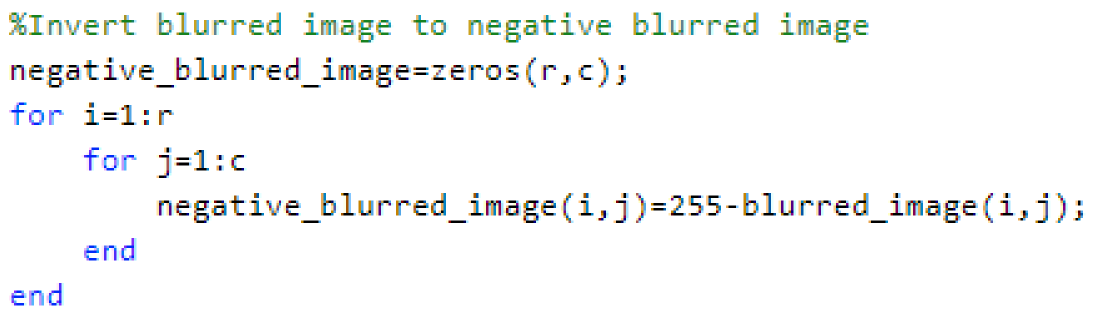
Then we blur the negative grayscale image with the gaussian method. First, we enter the sigma value. Then we enter the filter value. By using the fspecial command, we create a filter with the values we just entered. In the fspecial command, we choose the method we will use at first, namely gaussian, then we enter the filter and sigma value. As sigma increases, the image becomes more blurred. Sigma controls the extent of the variance and thus, the degree of blurring.



```
%Convert Negative Grayscale Image to Blurred Image
sigma = 10;
filter = 7 * sigma;
H = fspecial('gaussian',[filter,filter],sigma);
blurred_image = imfilter(negative_grayscale_image,H);
```

Figure 5 Negative grayscale to blurred image

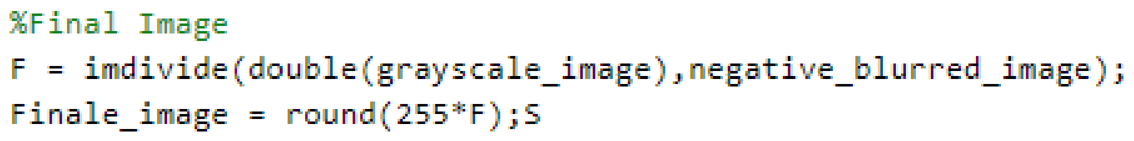
Then we find the negative of the blurred image. Here we use the same method that we got the negative of the grayscale image above.



```
%Invert blurred image to negative blurred image
negative_blurred_image=zeros(r,c);
for i=1:r
    for j=1:c
        negative_blurred_image(i,j)=255-blurred_image(i,j);
    end
end
```

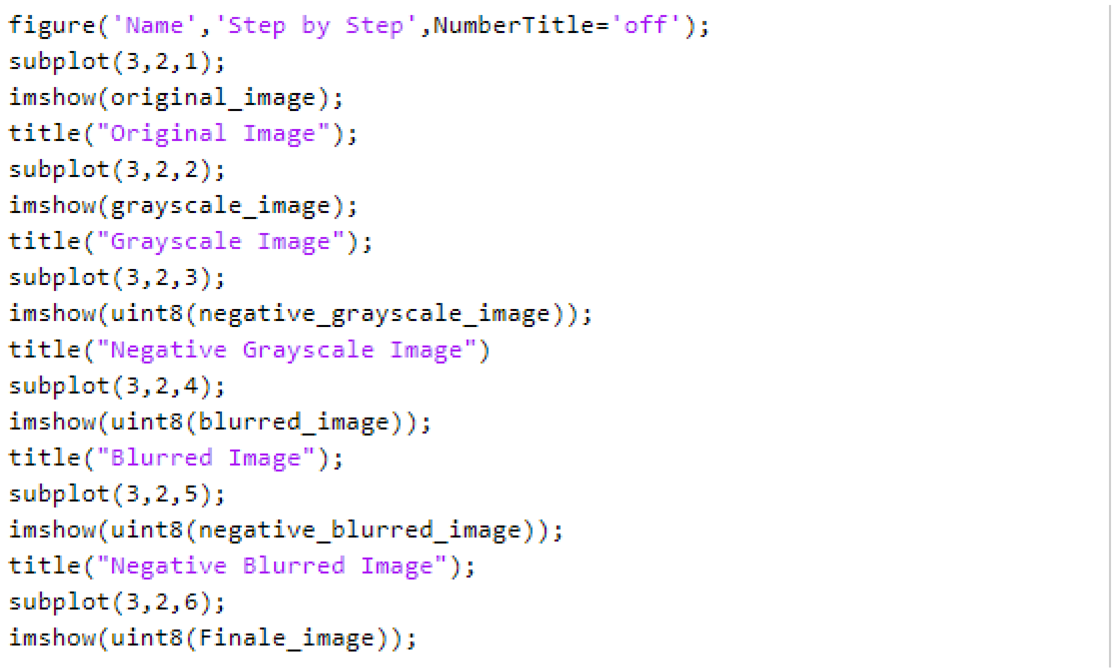
Figure 6 Blurred image to negative blurred image

The last step is to divide the grayscale image into a negative blur image. The sketch can be obtained by performing bit-wise division between the grayscale image and the inverted blurred image. Here, we multiply each pixel of the image we find by 255 and round it to the nearest integer with the round command.



```
%Final Image
F = imdivide(double(grayscale_image),negative_blurred_image);
Finale_image = round(255*F);S
```

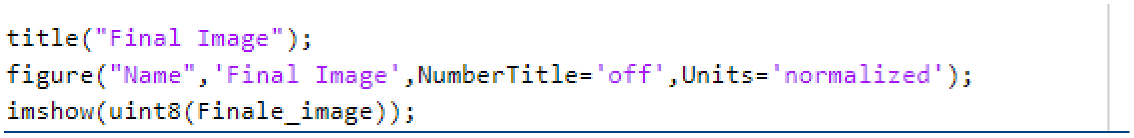
Figure 7 Final image



```
figure('Name','Step by Step',NumberTitle='off');
subplot(3,2,1);
imshow(original_image);
title("Original Image");
subplot(3,2,2);
imshow(grayscale_image);
title("Grayscale Image");
subplot(3,2,3);
imshow(uint8(negative_grayscale_image));
title("Negative Grayscale Image")
subplot(3,2,4);
imshow(uint8(blurred_image));
title("Blurred Image");
subplot(3,2,5);
imshow(uint8(negative_blurred_image));
title("Negative Blurred Image");
subplot(3,2,6);
imshow(uint8(Finale_image));
```

Figure 8 Figure-1 code

The second figure is printed on the screen in a larger way so that the resulting image can be examined in more detail.



```
title("Final Image");
figure("Name",'Final Image',NumberTitle='off',Units='normalized');
imshow(uint8(Finale_image));
```

Figure 9 Figure-2 code

In this program, we took an image of any desired size and processed it step by step and transformed the image as requested. Finally, I put the results of all the images I tried below.

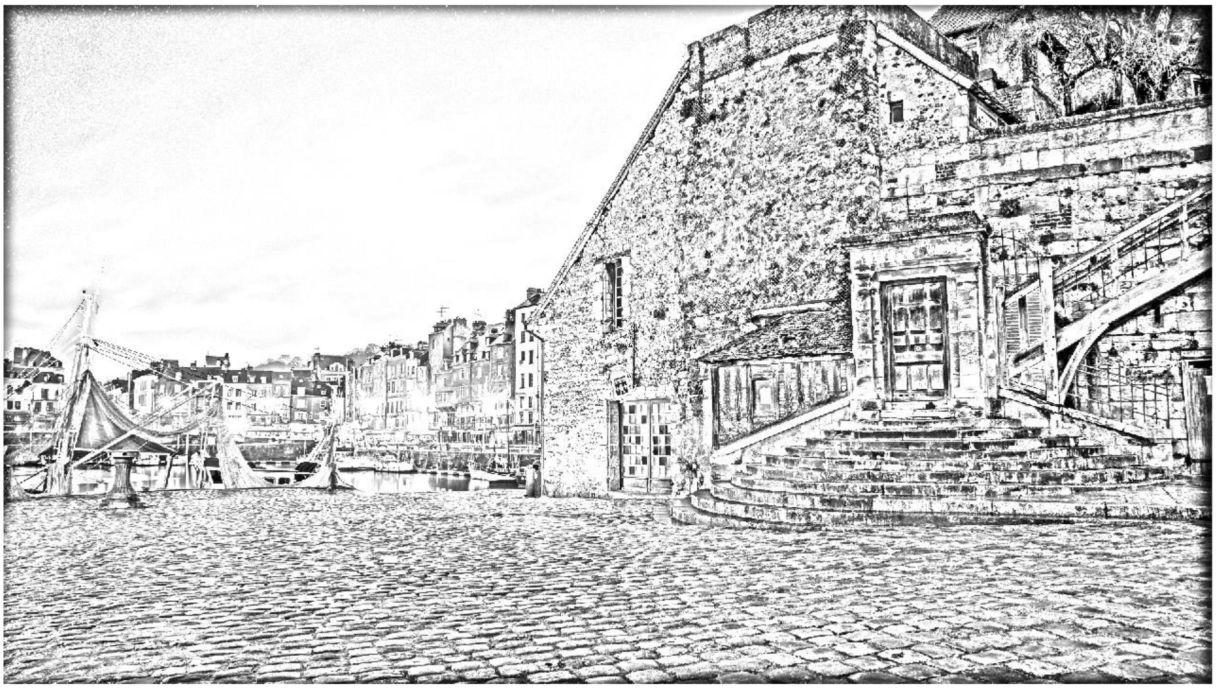


Figure 10 Example-1

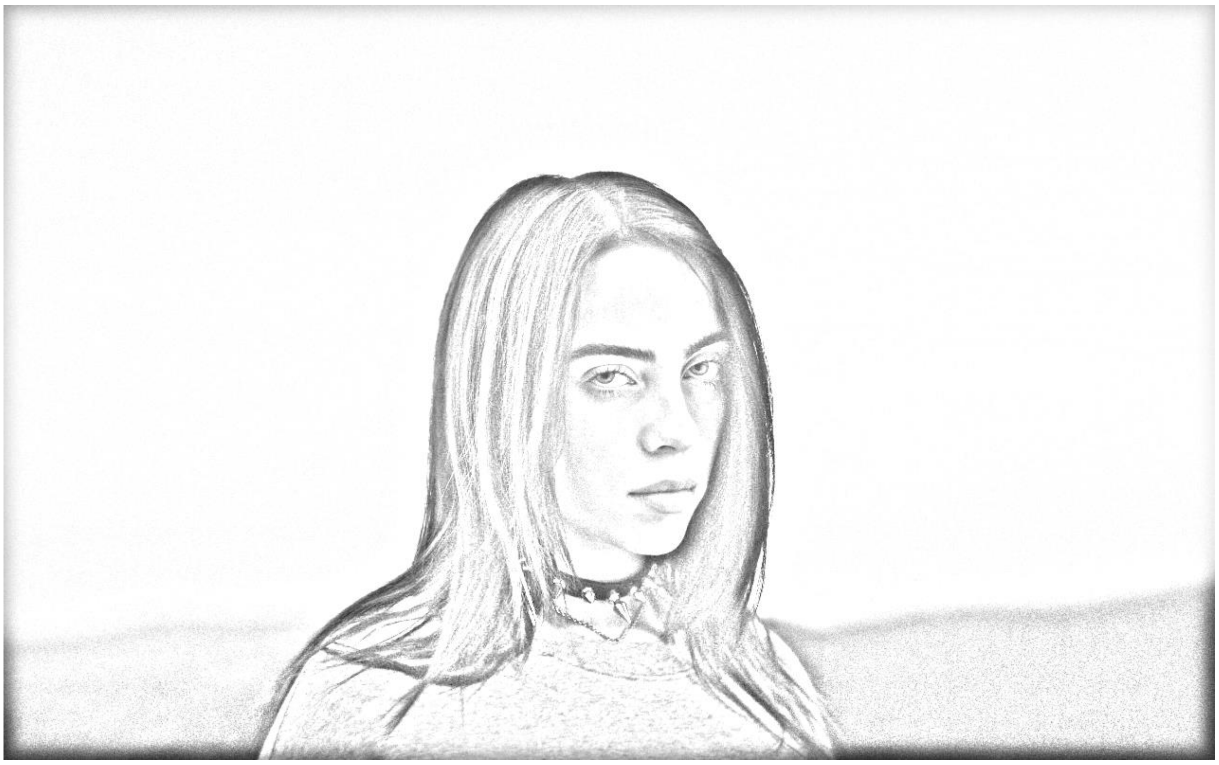


Figure 11 Example-2

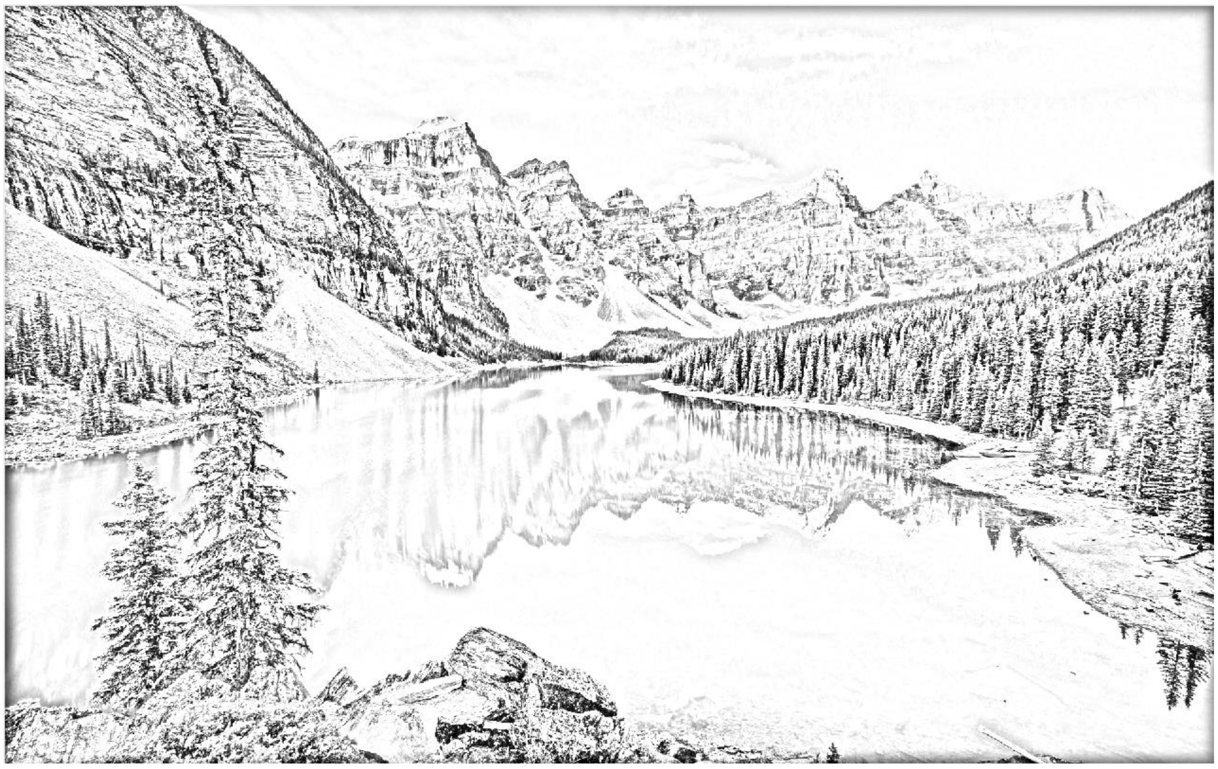


Figure 12 Example-3

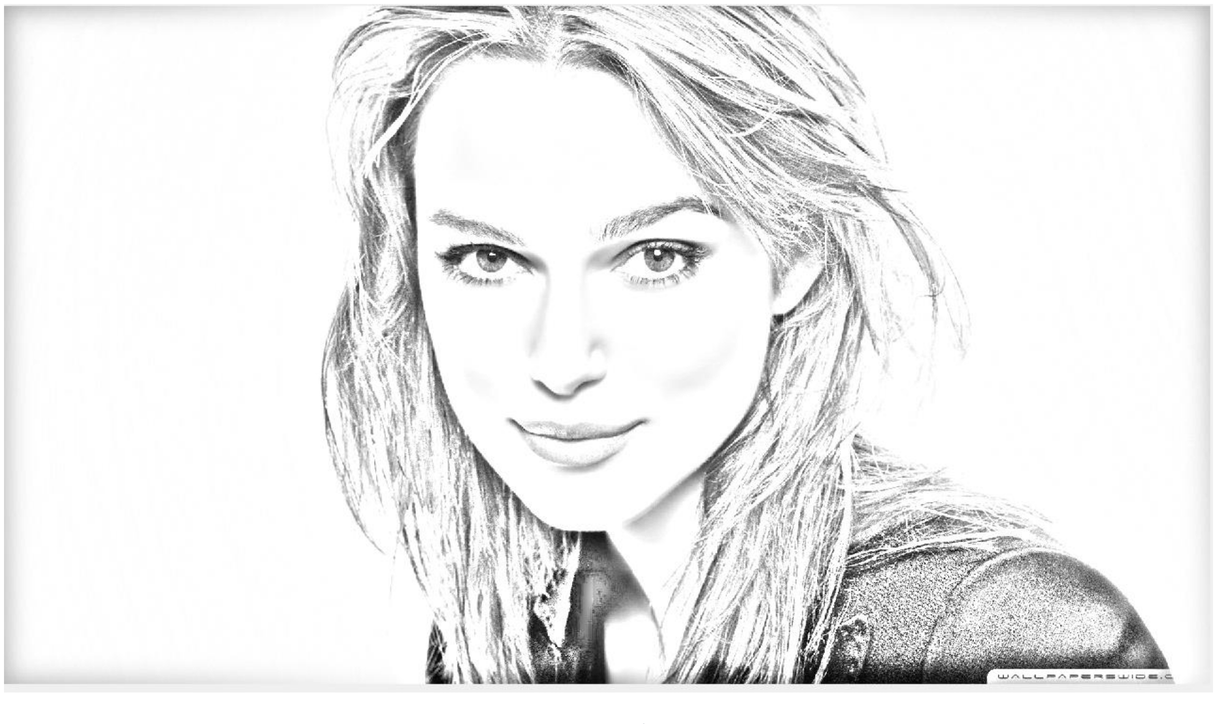


Figure 13 Example-4

References

- Image To Pencil Sketch In 12 Lines Of Code Using Python - YouTube
- Generate Pencil Sketch from Photo in Python | by Abhijith Chandradas | Towards Data Science
- Python: Converting Images to Pencil Sketch - AskPython
- How to turn any image into a pencil sketch with 10 lines of code (freecodecamp.org)