Nico Ferrari 514034

REPORT LAB 2

Task 1: Inverting input image into its negative.

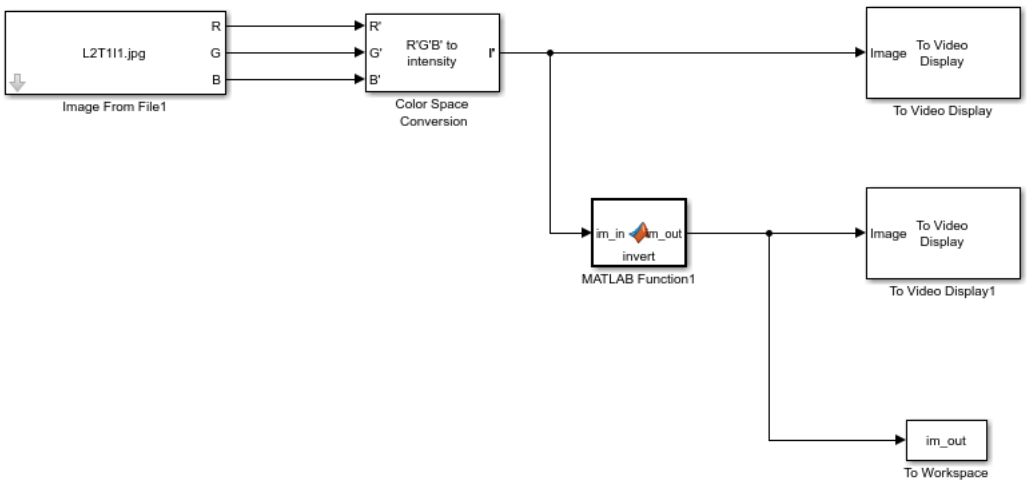
To invert a given image from the hard disk into its negative I used these blocks:

- Image from file: imports from the hard disk an image and outputs separate color signals. The sample period of the block is 1.
- Color space conversion: this block executes a color space conversion of the image in input from RGB color space to intensity.
- MATLAB function: executes 'invert' MATLAB function, that inverts an image into its negative and the algorithm is implemented as below

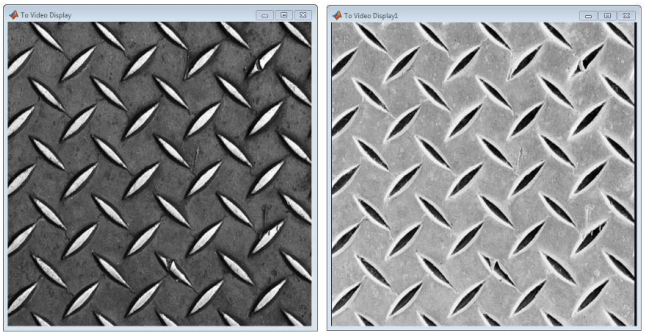
```
function im_out= invert(im_in)

im_out=imcomplement(im_in);
```

- Display block: to display the two images.
- To workspace block: to return the inverted image to MATLAB workspace.



This is the result:



On the left we have the image used as input and on the right the output.

We can notice that the MATLAB function performs a frame based processing, maximizing the efficiency.

Task 2: Inverting input image from Workspace into its negative.

For the task 2, as requested, I implemented two callback functions:

1. *Task2_preproc:* enabled at the beginning of the model execution

```
im=imread('C:\Users\nife1600\Documents\MATLAB\slprj\L2T1I1.jpg');

im=rgb2gray(im);
im=imresize(im,[64 64]);
im1=reshape(im,1,4096);
simin.time=(0:4095)';
simin.signals=struct('values',im1');
```

2. *Task2_postproc*: enabled at the end of the model execution

```
im2=reshape(simout,[64 64]);
subplot(1, 2, 1);
imshow(im)
subplot(1, 2, 2);
imshow(im2);
```

The block used for the model are:

- From Workspace: this block is used to get “simin” variable from the workspace, created during the preprocessing phase.
- MATLAB Function: this block is the same as task 1;
- To Workspace: this block is used to outputs the variable “simout”, that is the result of the image inversion, into the workspace.

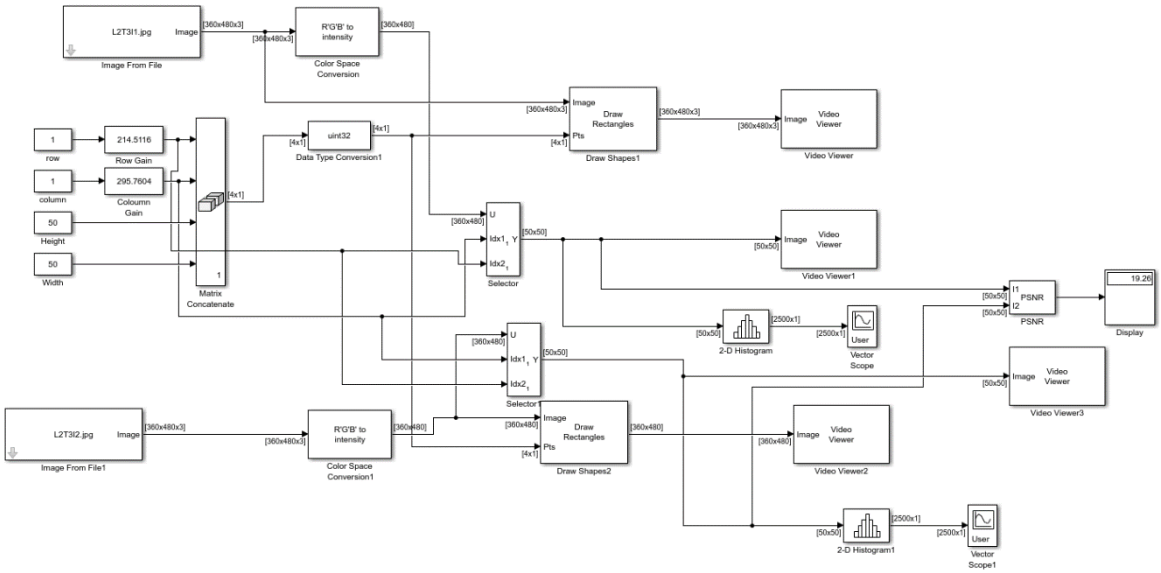
In this task, the MATLAB function performs a pixel based processing, minimizing the efficiency.

Task 3

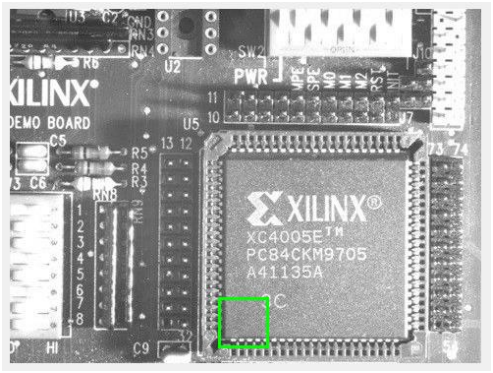
In the given model, there were some mistakes:

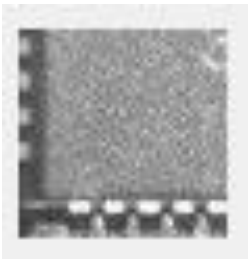
- “Row Gain” and “Column Gain” blocks have a wrong configuration and this mistake prevents us to select the portion of the image.
- The configuration of “Row Gain” and “Column Gain” results inverted.
- Row and columns limits must be decreased by the size of the selecting rectangle, to prevent the rectangle goes outside of the given image.

This is the final model:



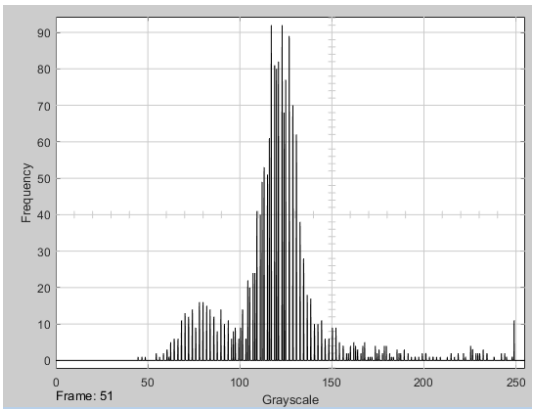
The first part of the model reads an image from file and marks a region as ROI with a green rectangle which can be positioned setting the Row Gain and Column Gain parameters.

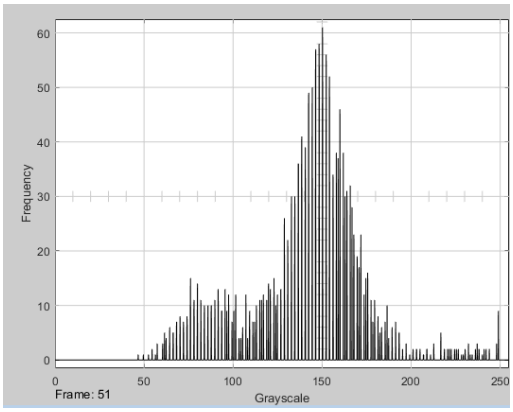




The second part plot the histogram for the selected region. It consists of:

- “2-D Histogram” block: produces the histogram of the selected region;
- “Vector Scope” block: to display the histogram.





The last part display the SNR of two images and consists of PSNR block that computes the peak signal-to-noise ratio (PSNR), in decibels, between the images.