

EE3025 IDP: Implementing Sobel Filter on Ico Board

Pathapati Aravind Ganesh

EE16BTECH11026

Bhavsar Jeel Alpesh

ES16BTECH11005

1 Introduction

Sobel filters are used in image processing to detect the edges of an image. We are applying a 8-bit, 3x3 sobel filter on 8-bit, 28x28 images with vertical and horizontal stride as 1 to receive a resultant 8 bit, 26x26 image.

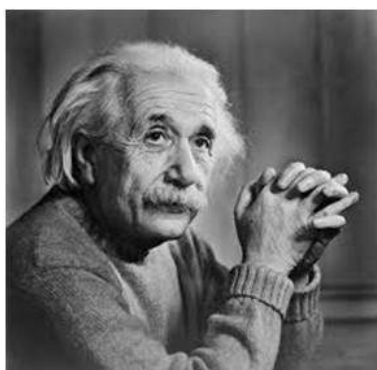


Figure 1: original image

1.1 Vertical mask of Sobel operator

When we convolve the image in 2D with the following mask, it will highlight the vertical edges of the image.

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 2 | 0 | -2 |
| 1 | 0 | -1 |

The above mask calculates the difference of pixel intensities in an edge region. Since the centre column is zero, the original values of the image are not included but rather it calculates the difference of right and left pixel values around that edge. The centre values of first and third column are 2 and -2 respectively to increase the weight age to pixel values around the edge region and enhance the edge.

1.2 Horizontal mask of Sobel operator

When we convolve the image in 2D with the following mask, it will highlight the horizontal edges



Figure 2: image after applying vertical mask

of the image.

| | | |
|----|----|----|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

The above mask works on the same principle as the vertical mask. Since the centre row is zero, the original values of the image are not included but rather it calculates the difference of top and bottom pixel values around that edge. The centre values of first and third row are 2 and -2 respectively to increase the weight age to pixel values around the edge region and enhance the edge.



Figure 3: image after applying horizontal mask

2 Implementation

We read the image in python in `image_initialization.py` and convert the pixel values into 8 bit binary format as required by verilog. The 28x28 image has 784 elements. After getting it into the correct format, the 784 values are initialized in `sobel_uart.v` automatically once you run the python code. Our codes are available at https://github.com/AravindGanesh/fpga_idp. The sobel filter(vertical or horizontal) is initialized in `sobel_uart.v` and the convolution takes place in `conv.v`.

Run the following commands in your terminal:

```
$make initialize  
$make vf_name=sobel_uart  
$make receive
```

It performs the operation on image chosen in `image_initialization.py` and saves the output image as a JPEG file.

We need to send the output 26x26 image from the icoboard back to the Raspberry Pi. The icoboard is connected to the GPIO pins so they are not accessible to us. Instead we use an FTDI to send the image back 1 pixel intensity(8 bit wide) at a time using UART protocol <https://github.com/ZipCPU/icozip/tree/master/rtl/uart>. Now we read the values received using `receive_img.py` and save the output image on the pi.

References

https://www.tutorialspoint.com/dip/sobel_operator.htm

https://github.com/AravindGanesh/fpga_idp/tree/master/src/sobel_filter