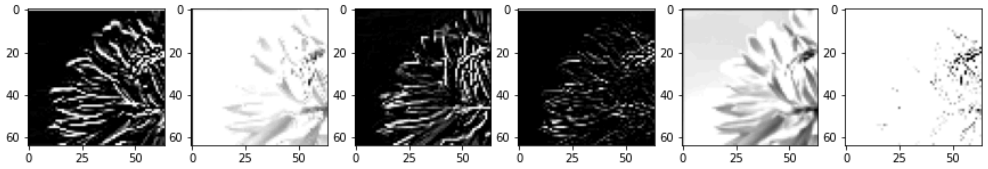
EECE 4632: Spring 2022
Hardware-Software Codesign for FPGA-Based Systems
Early Software Implementation
Oscar Kellner

Within this submission contains a Jupyter Notebook containing the code for the software implementation of the single convolution layer. The code takes an image (that must have the same width and height) and processes it into a defined number of output feature maps. Each feature map corresponds to a filter (kernel) defined within the code, of which there are 6. The convolution algorithm is also timed in seconds so that it can be compared later on once the hardware version is implemented.

```
In [146]: # Timing of software implementation
          start = time.time()
          output = conv(data2)
          # Normally a nonlinear operation is applied here, however for now the focus is
          # on the convolution layer algorithm itself
          #numpy_out = relu(data2)
          #numpy_out = sigmoid(data2)
          end = time.time()
          #print(numpy_out)
          print("Time: ", end-start)

          Time:  0.45284080505371094
```

```
In [147]: # Displaying the output feature maps
          fig, axi = plt.subplots(nrows=1, ncols=c_out, figsize=(15, 4))
          for i, ax in enumerate(axi.flatten()):
              plt.sca(ax)
              im = Image.fromarray(output[0][i]*255)
              im = im.convert("L")
              plt.imshow(im, cmap='gray')
          plt.show()
```



The next step is to isolate the code within the Darknet framework responsible for the convolution algorithm (or rewrite it in a way that makes it easier to use on HLS). The convolution algorithm is rife with nested loops, so there is a lot to explore with different optimizations that can be applied to the algorithm, however it is important to first focus towards having a working product before complex optimizations can be considered.

The code and image used to create the Jupyter Notebook is a heavily modified version of code found within an educational article, linked below.

Convolutional layer hacking with Python and Numpy, Javier Ideami, 2021 Mar 15
https://towardsdatascience.com/convolutional-layer-hacking-with-python-and-numpy-e5f64812ca0c