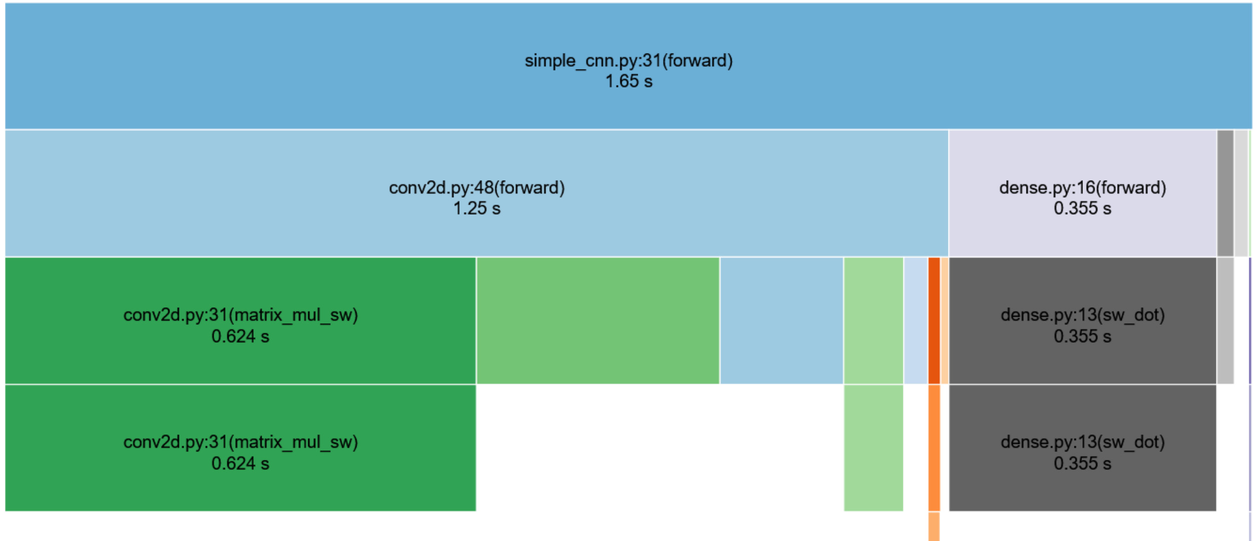
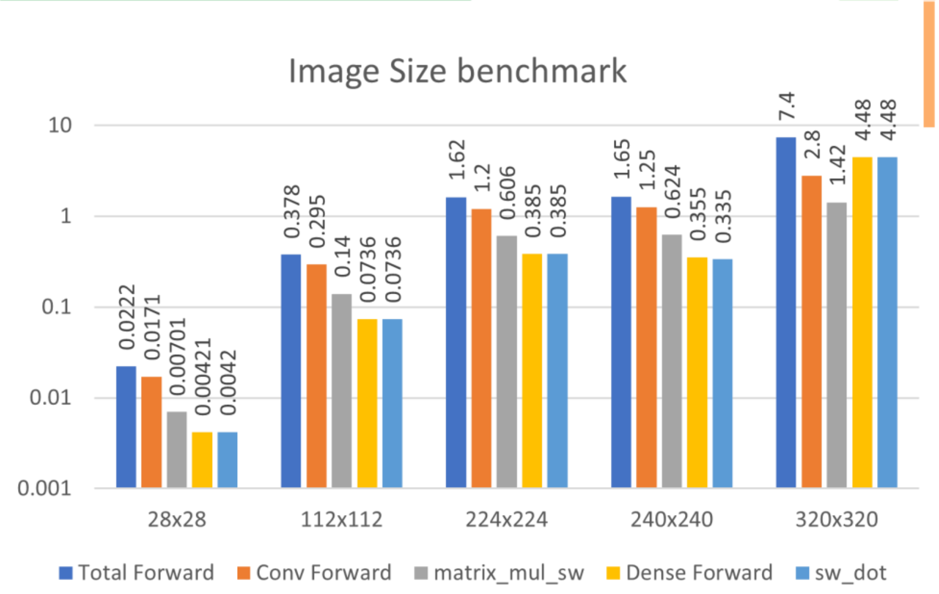
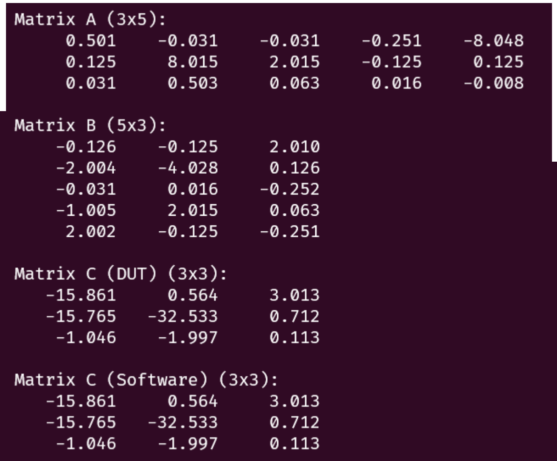
CNN Digit Recognizer for 240x240 Image

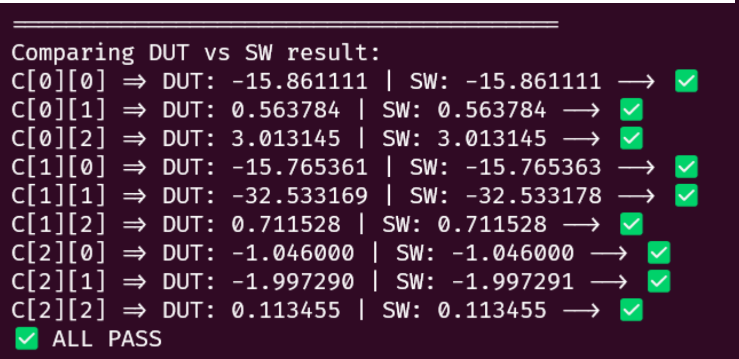
- *What are you trying to do?*
  - Build a system which can accelerate the convolution layers of the CNN
- *How have others implemented and/or accelerated this algorithm?*
  - Others use a 28x28 images from MNIST dataset
- *What are you doing differently/better/etc.?*
  - I am trying to do this for 240x240 images.
  - I have generated my own dataset for training.
- *What have you accomplished so far?*
  - Built the CNN from scratch using NumPy (no TensorFlow/PyTorch) and profiled it. (Numbers are in the images)
  - Flattened the convolution operation to Matrix Mul so it is easy to accelerate.
  - Built HW for Floating point Matrix Mul unit in Verilog and simulated it.
  - Fully function IEEE-754 32-bit precision Floating Point MAC unit.
  - Built a controller FSM for sequentially feeding each MAC operation.
- *What will you do next and what remains to be done?*
  - Benchmark the HW and take it through OpenLane flow.
  - Make the Floating point Matrix Mul unit parallel.
  - Connect HW and SW together (FPGA)
  - Experiment with precision.



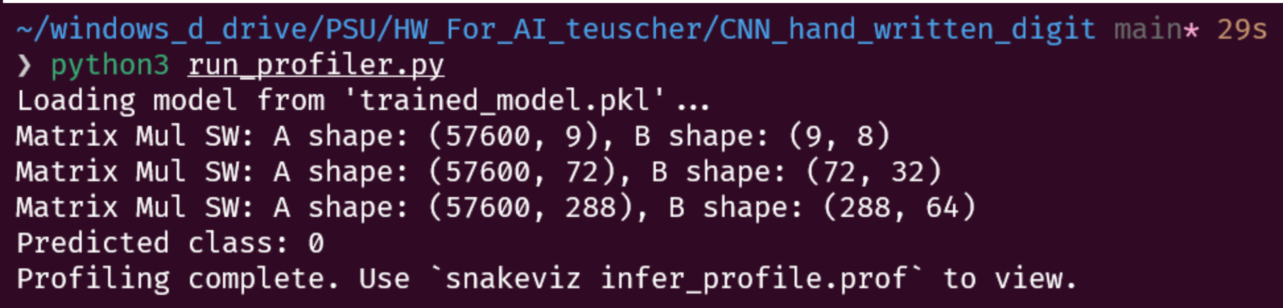




```
Matrix A (3x5):
      0.501    -0.031    -0.031    -0.251    -8.048
      0.125     8.015     2.015    -0.125     0.125
      0.031     0.503     0.063     0.016    -0.008

Matrix B (5x3):
     -0.126    -0.125     2.010
     -2.004    -4.028     0.126
     -0.031     0.016    -0.252
     -1.005     2.015     0.063
      2.002    -0.125    -0.251

Matrix C (DUT) (3x3):
    -15.861     0.564     3.013
    -15.765   -32.533     0.712
     -1.046    -1.997     0.113

Matrix C (Software) (3x3):
    -15.861     0.564     3.013
    -15.765   -32.533     0.712
     -1.046    -1.997     0.113
```



```
════════════════════════════════════════
Comparing DUT vs SW result:
C[0][0] ⇒ DUT: -15.861111 | SW: -15.861111 ⟶ ✅
C[0][1] ⇒ DUT: 0.563784 | SW: 0.563784 ⟶ ✅
C[0][2] ⇒ DUT: 3.013145 | SW: 3.013145 ⟶ ✅
C[1][0] ⇒ DUT: -15.765361 | SW: -15.765363 ⟶ ✅
C[1][1] ⇒ DUT: -32.533169 | SW: -32.533178 ⟶ ✅
C[1][2] ⇒ DUT: 0.711528 | SW: 0.711528 ⟶ ✅
C[2][0] ⇒ DUT: -1.046000 | SW: -1.046000 ⟶ ✅
C[2][1] ⇒ DUT: -1.997290 | SW: -1.997291 ⟶ ✅
C[2][2] ⇒ DUT: 0.113455 | SW: 0.113455 ⟶ ✅
✅ ALL PASS
```



```
~/windows_d_drive/PSU/HW_For_AI_teuscher/CNN_hand_written_digit main* 29s
❯ python3 run_profiler.py
Loading model from 'trained_model.pkl'...
Matrix Mul SW: A shape: (57600, 9), B shape: (9, 8)
Matrix Mul SW: A shape: (57600, 72), B shape: (72, 32)
Matrix Mul SW: A shape: (57600, 288), B shape: (288, 64)
Predicted class: 0
Profiling complete. Use `snakeviz infer_profile.prof` to view.
```