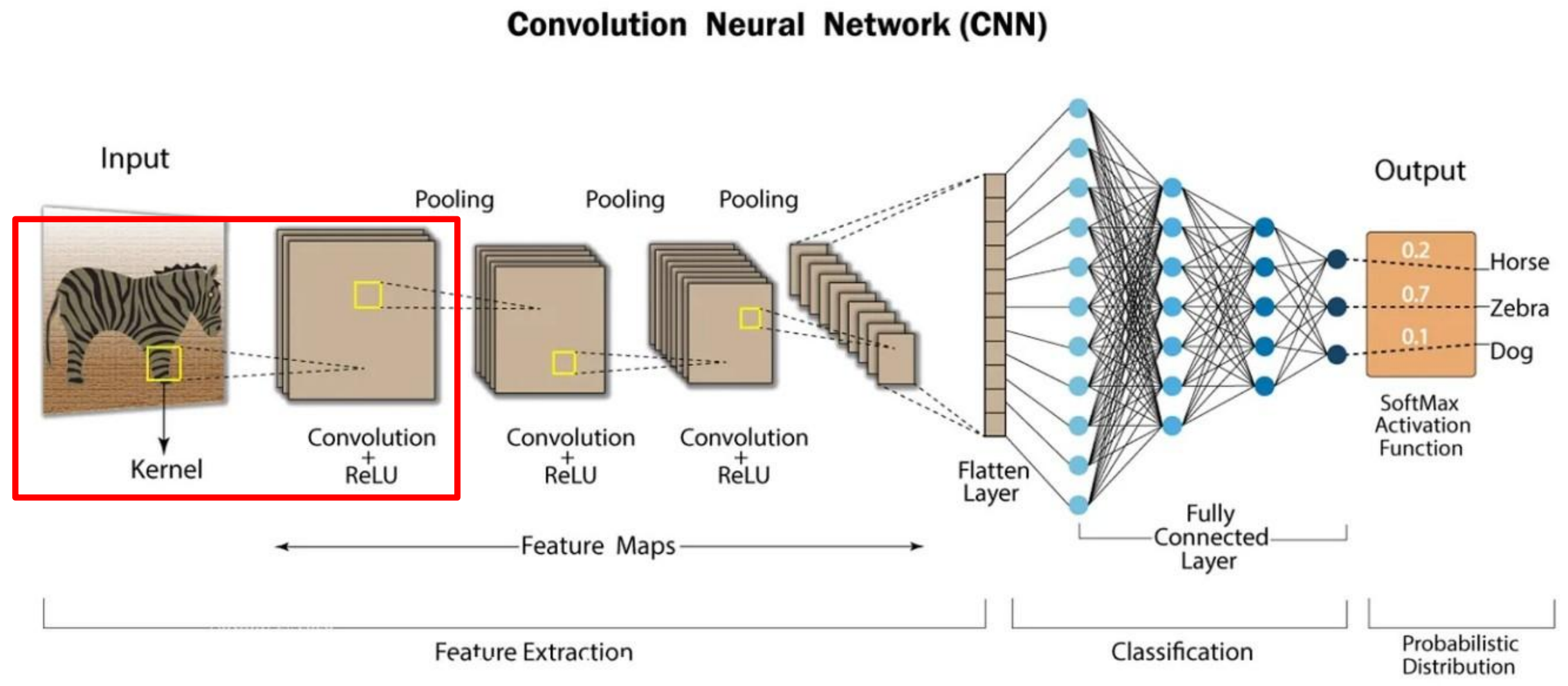
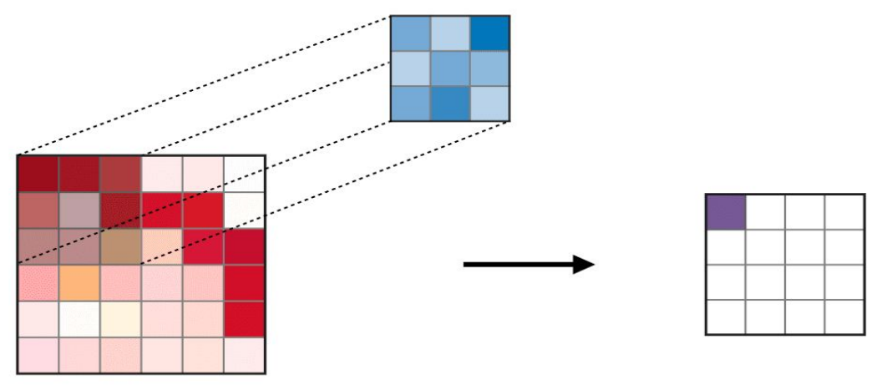
Hardware acceleration of Computer Vision Algorithms using Field Programmable Gate Arrays (FPGA)

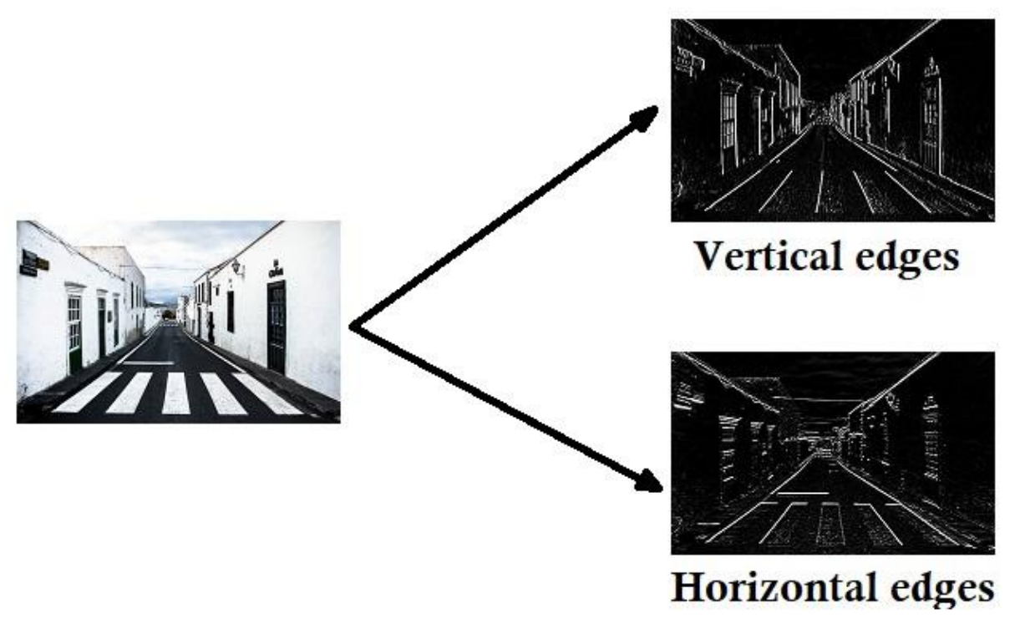
Author(s): Dengyu(Spud) Tu Advisor: Dr. Hunter Adams

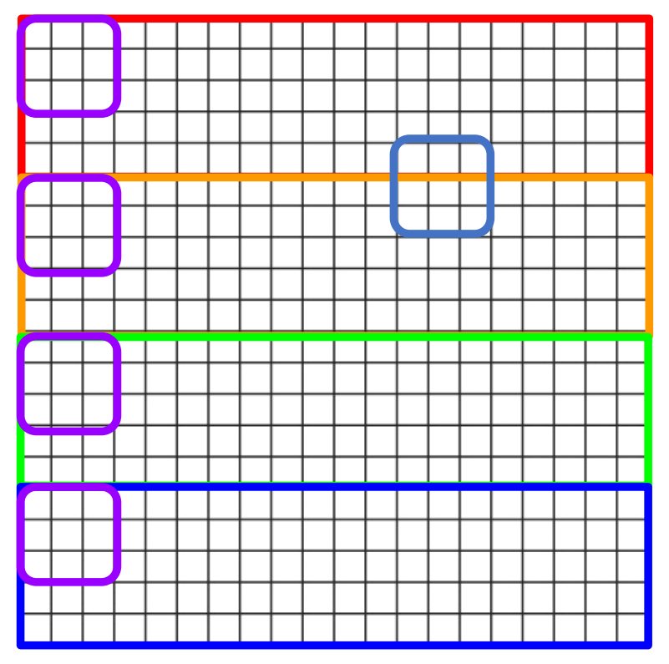
Why is faster CNN computation critical?

- Computer vision algorithms are widely used in technologies like Face ID, Autopilot, and Amazon Go, etc.
- Convolutional Neural Networks (CNNs) serve as the foundational framework behind these successful applications.
- The motivation of this project is to enhance the processing speed of CNNs to meet the rising demand for high-speed automation.
- CNNs consist of multiple layers of calculations, with the focus of this project being on the first two layers.
- Convolution operations in CNNs involve sweeping a kernel across input images, generating different results based on the kernel used.
- The uniformity of kernels across CNN layers allows for straightforward parallelization, making FPGA acceleration an efficient solution.

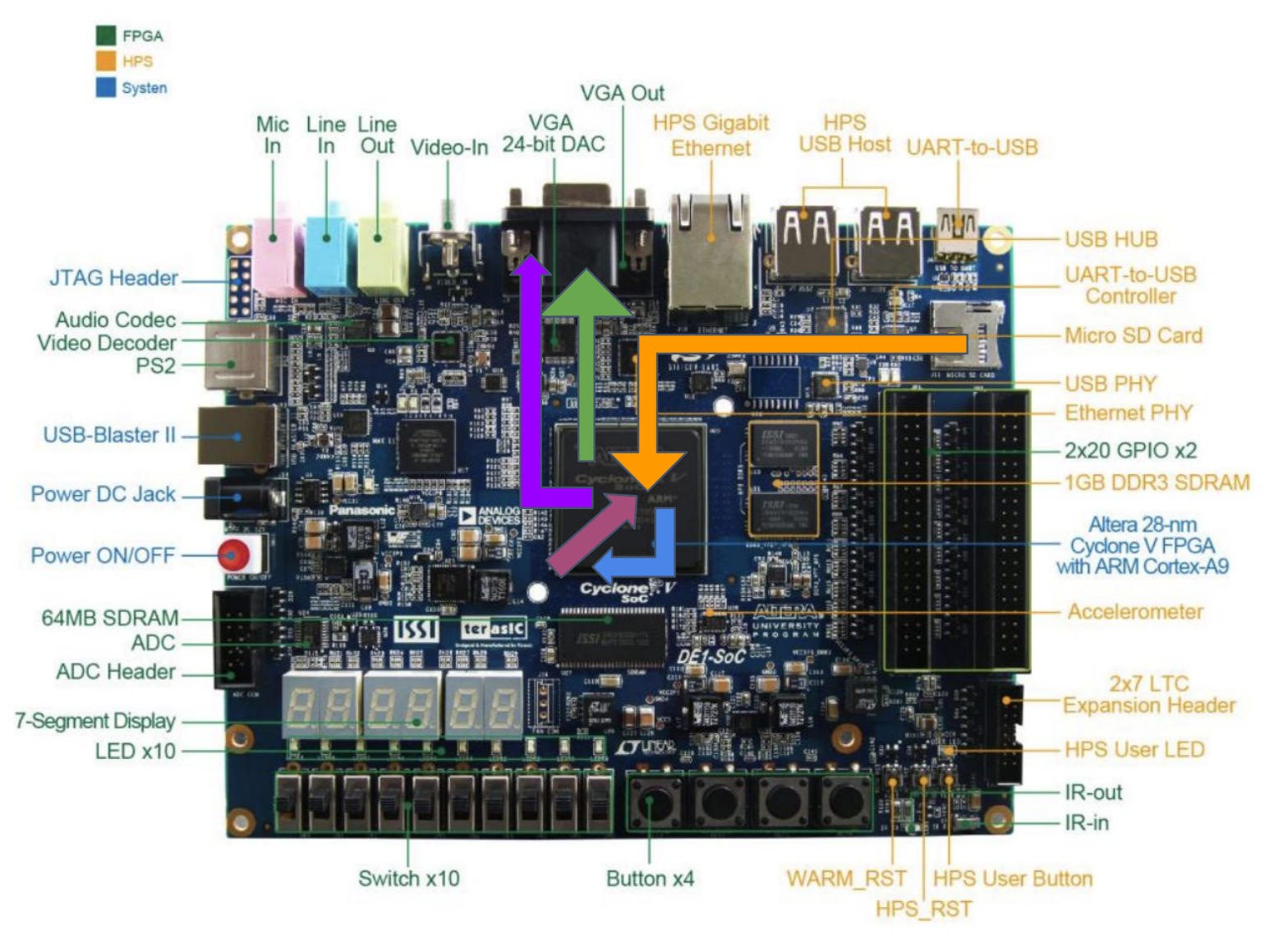


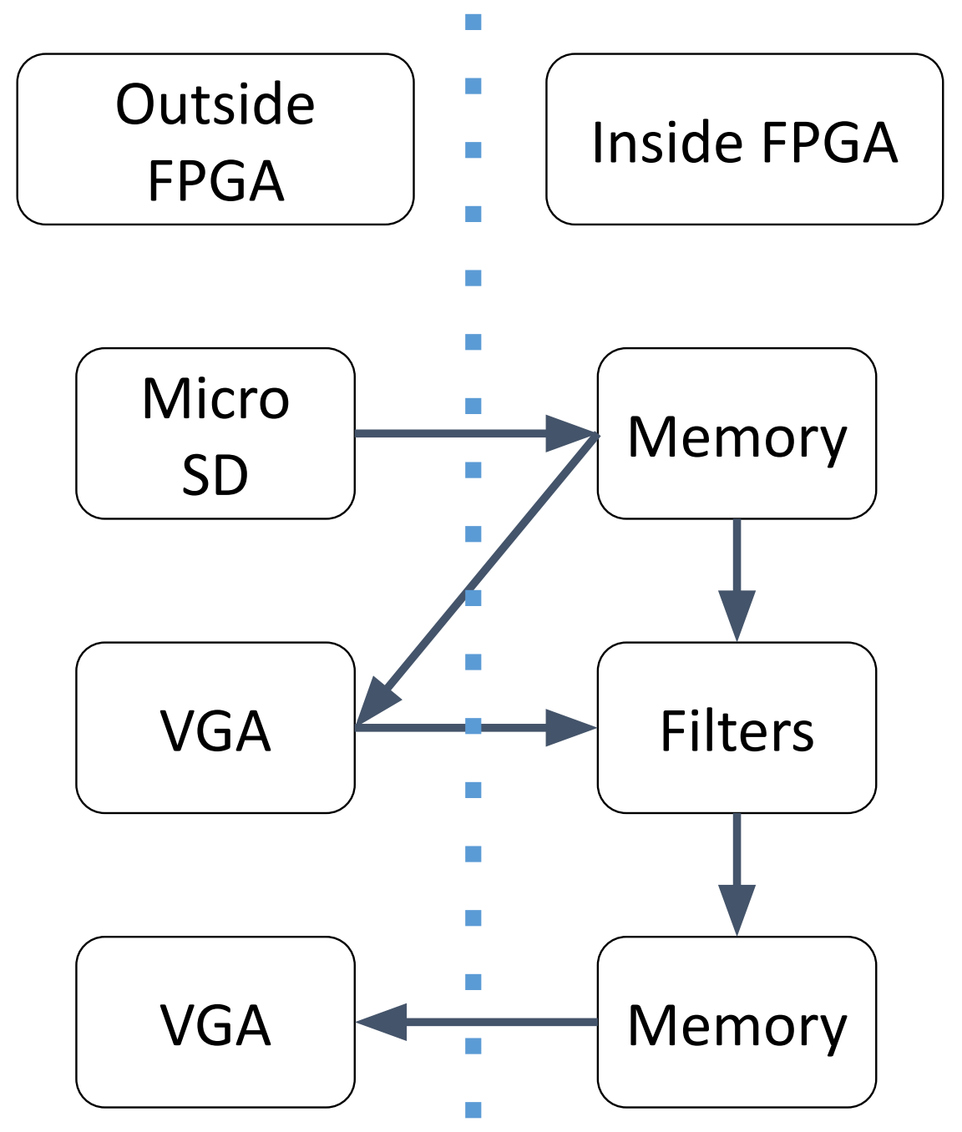


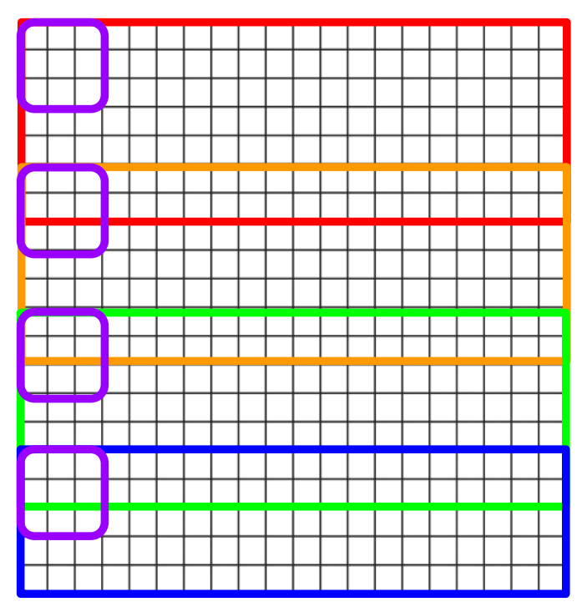


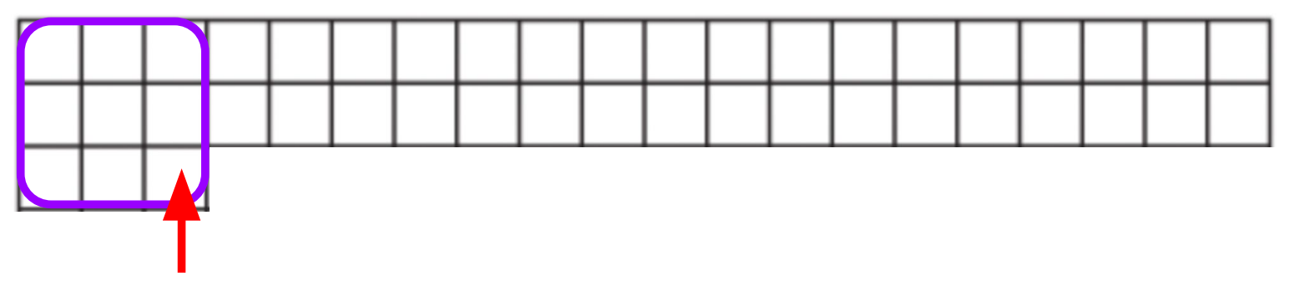


How to implement this algorithm?









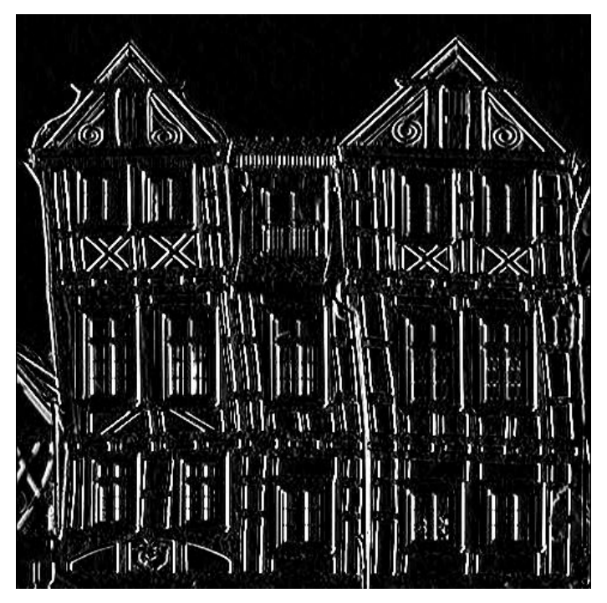
Next Input

- The pixel data first flows from Micro SD card to the M10k memory in the FPGA. And the VGA driver reads data directly from the M10k and displays on the VGA screen. The filter is then convoluted with the data in the M10k and stores them back after processing. Lastly, the VGA driver displays the updated pixel information on VGA screen.
- To optimize memory access in light of the M10k memory's restriction of reading only one data per cycle, a buffering mechanism was implemented. This involved initially reading a segment of the M10k memory into a buffer, allowing for pipelining of the kernel calculation with memory reads.
- Each pixel value is represented by 8 bits, where the most significant three bits correspond to the red color channel, the middle three bits to the green color channel, and the last two bits to the blue color channel.
- To calculate the pixel at the edges of different paralyzed sections, each M10k blocks overlap with others by two rows.
- All RGB values are constrained within the range of 0 to 255 to maintain color consistency and prevent data overflow.
- Zero padding was applied to the edges of the images to preserve its dimension.

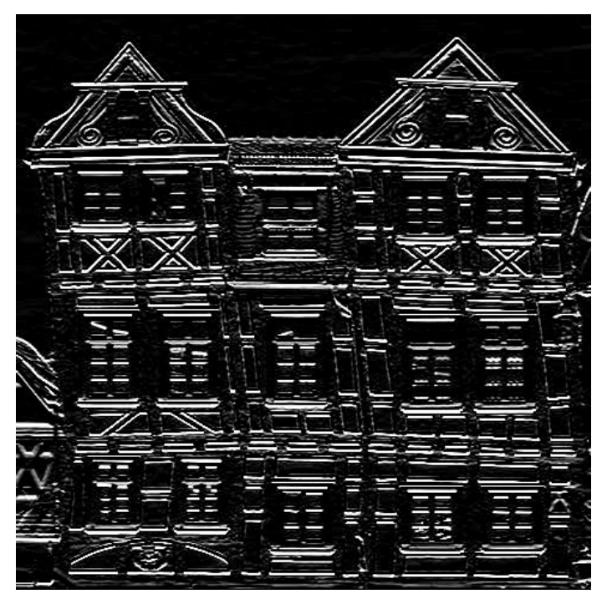
CornellEngineering
Electrical and Computer Engineering

How well does the FPGA work?

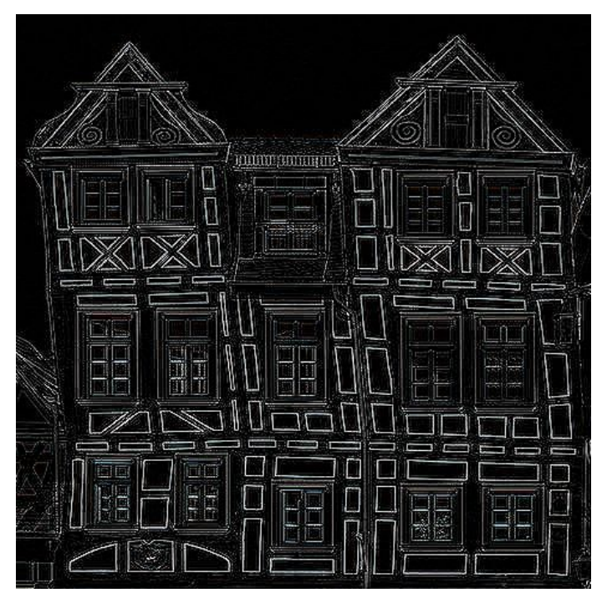
1	0	-1
2	0	-2
1	0	-1

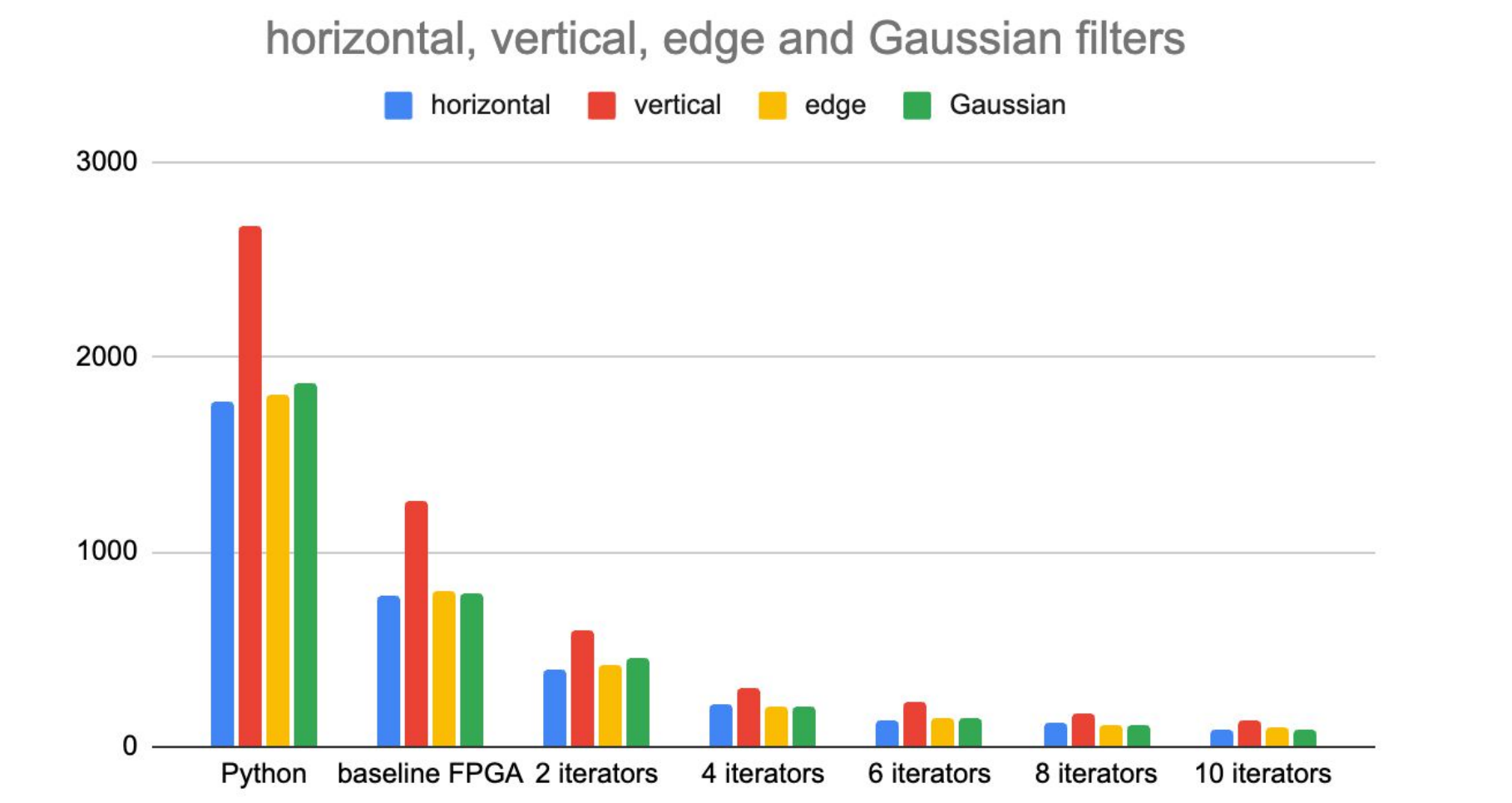


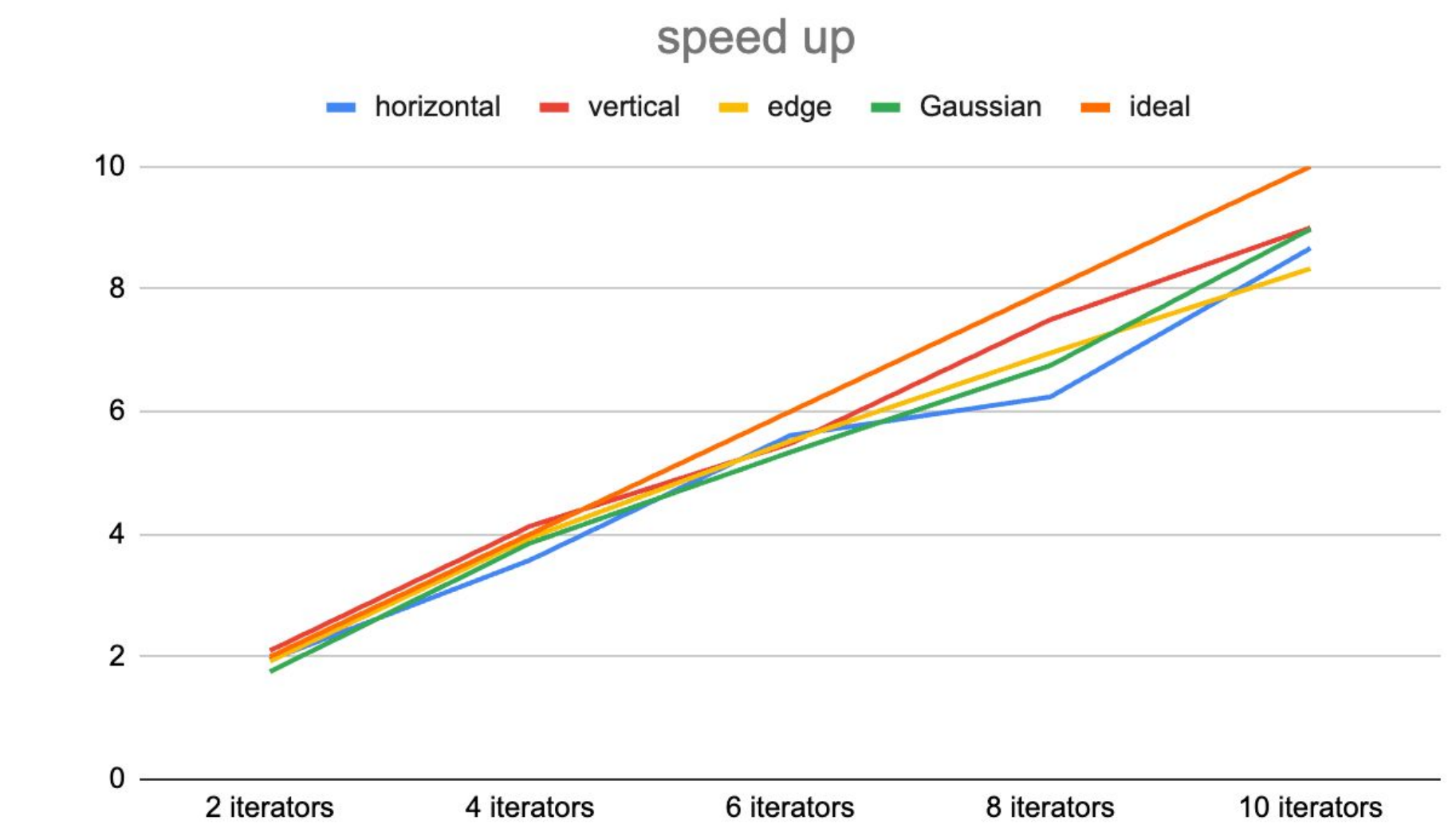
1	2	1
0	0	0
-1	-2	-1



0	-1	0
-1	4	-1
0	-1	0







- All filters exhibit distinct results, affirming the accuracy of the convolution layer calculations.
- Increased kernel iterations and parallelization accelerate the computation speed.
- The more iterators we use, the more registers we need which creates a trade of between efficiency and FPGA usage.
- Parallelization yields notable speed enhancements, with timing changes nearly linearly correlated to hardware iteration counts

What do we learn?

- The FPGA outperforms the CPU in calculations, validating the idea of accelerating computer vision algorithms.
- As shown in the bar plot, implementing more parallelism and parallelized kernels results in faster hardware acceleration.
- Even with an input buffer, the overall speed is still constraint by the read speed of M10k memory. If we can read memory faster or read more data each clock cycle, we can pipeline the algorithms more effectively and achieve a significant speedup.

If time permits…

- We can measure the power consumption of FPGA which is another huge benefit using FPGA for hardware acceleration.
- We can have more comparison between FPGA versus CPU and FPGA versus GPU.
- We can assess whether FPGA offers superior energy efficiency compared to GPU.
- We can implement the complete Convolutional Neural Network including the Relu function, Pooling layer and Fully Connected Layers.

Acknowledgement

I would like to thank my advisor Dr. Hunter Adams for his advice, encouragement, and continued support of this project.

Reference

- https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9
- https://cs231n.github.io/convolutional-networks/
- https://www.linkedin.com/pulse/what-convolutional-neural-network-cnn-deep-learning-nafiz-shahriar/
- https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/DE1-SoC_User_manualv.1.2.2_revE.pdf
- https://www.augmentedstartups.com/blog/the-best-object-detection-methods-for-2023-a-comprehensive-guide
- https://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/s2023/cp444_xz598/cp444_xz598/index.html