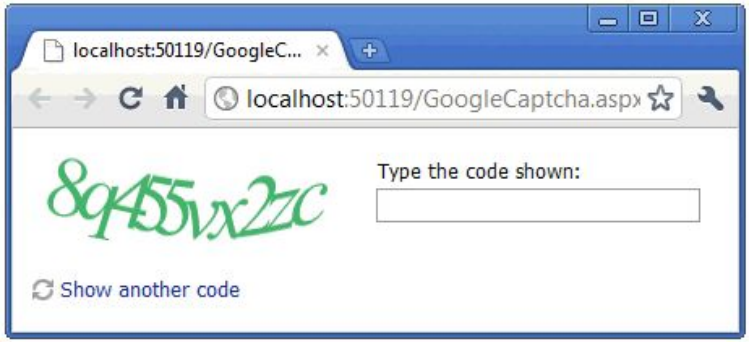
The Captcha Beater

Mauris, Nikhil, Randy & Srivatsan

Project Overview

- We will focus on text recognition for Captchas
- Implement a neural network on an FPGA to tackle the issue
- Potential applications:
  - Malicious captcha farms
  - Captcha solving assistant to individuals that are visually impaired
  - Converting handwriting to text
  - Finding vulnerabilities in captcha generators
  - Captcha avoidance for automated testing purposes



Challenges

- Finding a good quality dataset that is readily available
  - Possibly making our own dataset to circumvent this issue
- Coming up with an appropriate neural network structure
  - We need to consider hardware limitations
  - Should we used a “pre-trained” NN such as AlexNet and modify it?
- How do we handle the input images?
  - Do we need to compress?
    - On the FPGA or in software?
- Fastest or most secure method of communicating with the FPGA?
- How do we train the model on the FPGA?

Previous Work

A CAPTCHA Recognition Technology Based on Deep Learning

- Contrast Normalization used to normalized the brightness of the pixels, biased by a certain constant.
- Multi Task Joint Training trains each task to identify a specific character and the whole NN is trained in parallel to maximize efficiency.
- VGG-net, NN based on Alex-Net, used to solve 5 letter CAPTCHAs.
- Results show the model has a 96.5% accuracy.

Y. Hu, L. Chen and J. Cheng, "A CAPTCHA recognition technology based on deep learning," *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Wuhan, 2018, pp. 617-620, doi: 10.1109/ICIEA.2018.8397789.
(https://ieeexplore.ieee.org/document/8397789)

Previous Work

CAPTCHA Recognition with Active Deep Learning

- Goal is to reduce training data size by using Active Deep Learning
- Train set formed by choosing most informative data - based on uncertainty algorithm.
- 4-Layer CNN using 62 output neurons for each character in a 5 letter CAPTCHA
- Results show that a small train set was able to achieve 80% accuracy using active deep learning vs. 60% accuracy when using traditional training

Stark, Fabian & Hazırbaş, Caner & Triebel, Rudolph & Cremers, Daniel. (2015). CAPTCHA Recognition with Active Deep Learning. (https://www.researchgate.net/publication/301620459_CAPTCHA_Recognition_with_Active_Deep_Learning)

Previous Work

A Multi-Label Neural Network Approach to Solving Connected CAPTCHAs

- Python Package “CAPTCHA 0.2.1” was used as part of training data
- Multi-Label Learning approach used to solve the whole image instead of each letter separately.
- Using separate adjacent layers for next layer input to represent character connections
- 4-Layer CNN used to classify 4 letter CAPTCHAs
- Results show 94.26% accuracy on “CAPTCHA 0.2.1” set

K. Qing and R. Zhang, "A Multi-Label Neural Network Approach to Solving Connected CAPTCHAs," *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, 2017, pp. 1313-1317, doi: 10.1109/ICDAR.2017.216. (https://ieeexplore-ieee-org.myaccess.library.utoronto.ca/document/8270147)

Previous Work

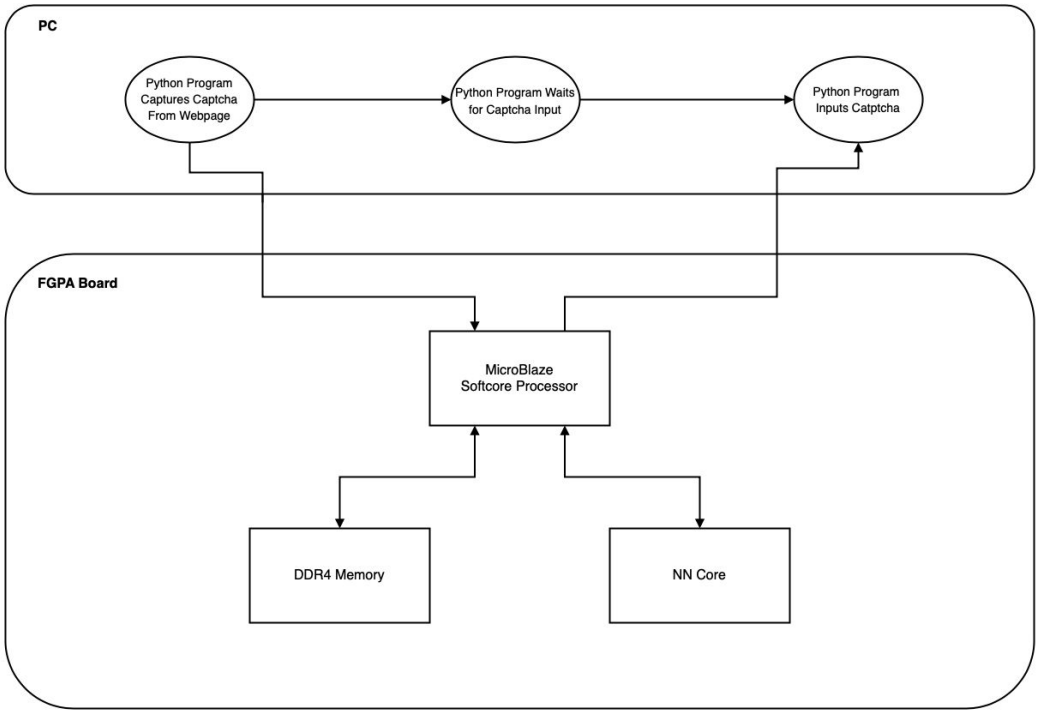
Verification CAPTCHA Based on Deep Learning

- Used PIL Library in Python to generate CAPTCHA images.
- The paper introduces concepts of CNN and Keras applied to CAPTCHAs
- Image Pre-Processed to remove color and split characters
- VGG-net based CNN used to solve 4-letter CAPTCHAs
- Results show 90% accuracy when solving CAPTCHAs with around 2 seconds of solve time

T. Zhang, H. Zheng and L. Zhang, "Verification CAPTCHA Based on Deep Learning," 2018 37th Chinese Control Conference (CCC), Wuhan, 2018, pp. 9056-9060, doi: 10.23919/ChiCC.2018.8482847.
(https://ieeexplore-ieee-org.myaccess.library.utoronto.ca/document/8482847)

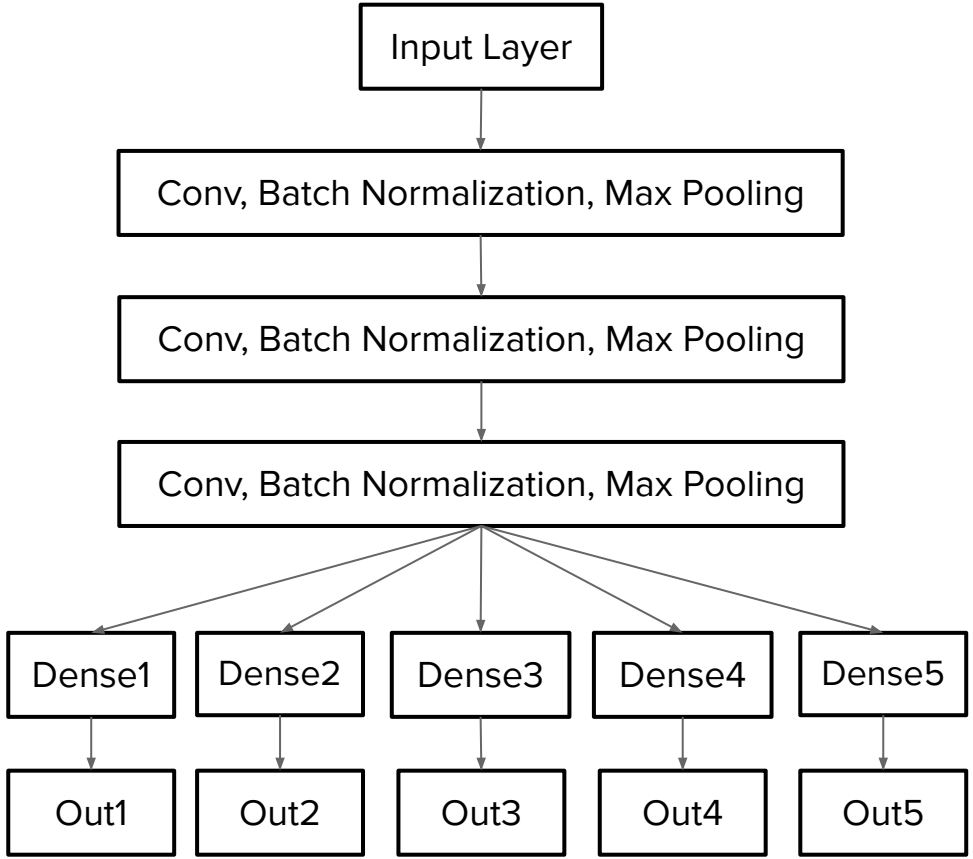
Proposed System

- Python program captures captcha and sends image to FPGA memory
- Softcore Microblaze conducts forward propagation to predict captcha
- Captcha prediction is sent back to PC and entered in
- Softcore MicroBlaze processor can be used to train neural network if time permits



Neural Net Design

- Planning on using a model that can splice each character in the captcha and classify them individually
- Model will have ‘n’ outputs for ‘n’ characters of the captcha
- Subset of characters to be used (19 total valid characters for now)



HLS Components

- Neural Network will be implemented in HLS
- What else is needed that is not HLS?
  - Softcore MicroBlaze
  - DDR4 Memory
  - Communication Components - Use FPGA IP Cores for data transfer between PCIe/UART/Ethernet, Memory, Microprocessor & Neural Network Core

Hardware Requirements

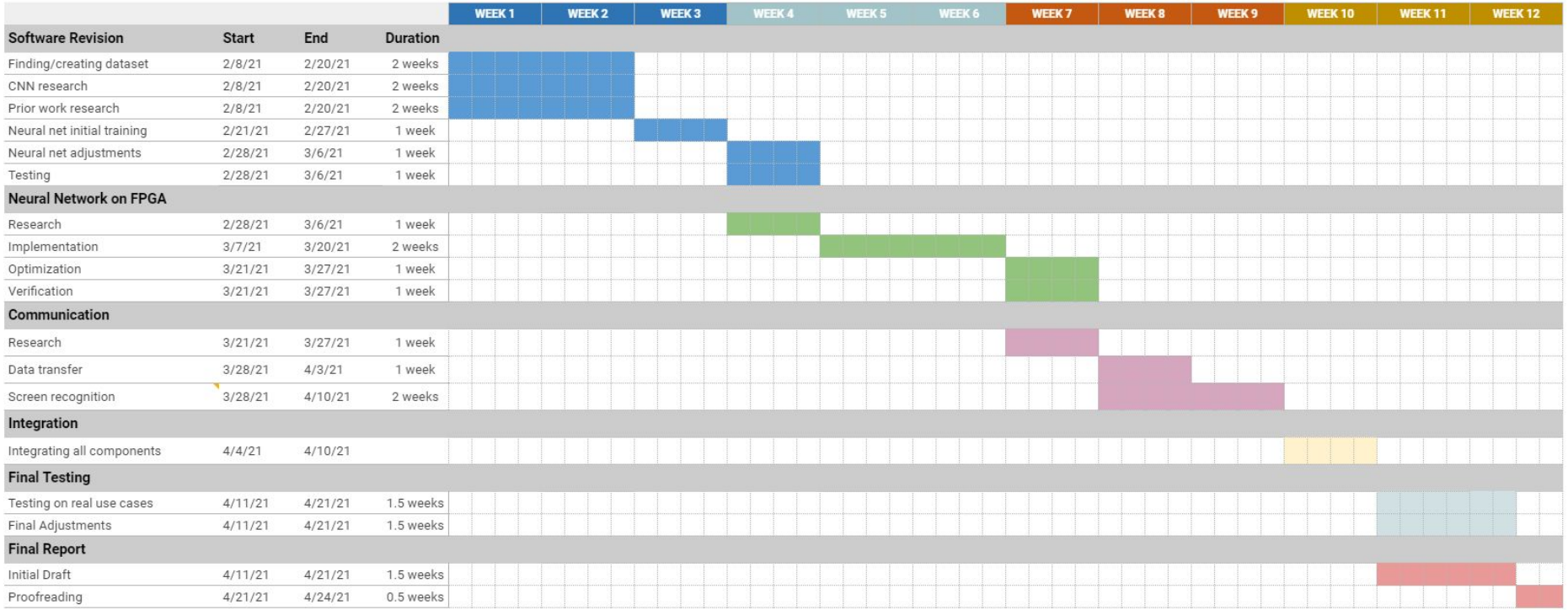
- FPGA Board - ADM-PCIE-8V3
  - Xilinx Virtex® UltraScale™ XCVU095-2 - FFVC1517
  - 2x 100G Ethernet MACs (incl. KR4 RS-FEC)
  - 2x 150G Interlaken cores
  - 4x PCI Express x8 Gen3

- PC & FPGA communication through PCIe Gen3? Ethernet? USB?
- Time permitting we would like to include multiple FPGAs to simulate a cloud infrastructure

Plan and Milestones

1. Get a software version working
    - Finding or creating an appropriate dataset
    - Using that dataset to train a neural network
    - Achieve reasonable results/accuracy
2. Get a neural net working on an FPGA through HLS
3. Developing a method to communicate with an external PC for data transfer (Images In, Results Out)
4. Integration
    - Send a picture to the FPGA
    - FPGA interprets picture and determines result
    - FPGA sends result back to PC
    - Make fixes as necessary
5. Test the application on a legitimate website or customized application
6. Complete the final report

Timeline (Ideal)



Software Revision	Start	End	Duration
Finding/creating dataset	2/8/21	2/20/21	2 weeks
CNN research	2/8/21	2/20/21	2 weeks
Prior work research	2/8/21	2/20/21	2 weeks
Neural net initial training	2/21/21	2/27/21	1 week
Neural net adjustments	2/28/21	3/6/21	1 week
Testing	2/28/21	3/6/21	1 week
Neural Network on FPGA			
Research	2/28/21	3/6/21	1 week
Implementation	3/7/21	3/20/21	2 weeks
Optimization	3/21/21	3/27/21	1 week
Verification	3/21/21	3/27/21	1 week
Communication			
Research	3/21/21	3/27/21	1 week
Data transfer	3/28/21	4/3/21	1 week
Screen recognition	3/28/21	4/10/21	2 weeks
Integration			
Integrating all components	4/4/21	4/10/21	
Final Testing			
Testing on real use cases	4/11/21	4/21/21	1.5 weeks
Final Adjustments	4/11/21	4/21/21	1.5 weeks
Final Report			
Initial Draft	4/11/21	4/21/21	1.5 weeks
Proofreading	4/21/21	4/24/21	0.5 weeks

Progress to Date

- Currently on track with planned timeline
- Developed initial machine learning model in software
- Currently tuning machine learning model and modifying training parameters to improve performance
- Currently constructing a more complete dataset