

CS CAPSTONE PROGRESS REPORT

JUNE 12, 2017

AI GAMING

PREPARED FOR

NVIDIA

MARK EBERSOLE

Signature

Date

PREPARED BY

GROUP 14

DEEP LEARNING ON EMBEDDED PLATFORM

CHRISTOPHER JOHNSON

Signature

Date

GABE MOREY

Signature

Date

LUAY ALSHAWI

Signature

Date

Abstract

In this document we describe everything we have done in our project to create a deep learning program that can learn to play the game Galaga. The project was created to be used as a training tool for NVIDIA's Deep Learning Institute. This document covers every piece of development material, including previously created documents, changes, and the final design of the project. Each team member will also present their logs of the development process and a summary of their learning throughout the term.

CONTENTS

1	Introduction	4
2	Original Documentation	4
3	Changes made from the requirements document	5
3.1	Change Table	5
3.2	Altered Gantt Chart	7
4	Changes made from the design document	7
5	Changes made from the technology review	8
6	Final Design	8
6.1	Context	8
6.2	Summary	9
6.3	Neural Network Setup	9
6.4	Communication	9
6.5	Hardware Setup	10
7	Guide	10
7.1	Installing Caffe	10
7.2	Set up image dataset	10
7.3	Training	10
7.4	Play the game	11
7.4.1	Game Hosting Computer	11
7.4.2	Jetson TX1	11
8	Learning Resources	11
9	Learning Reflection	11
9.1	Gabe's Reflection	11
9.2	Luay's Reflection	12
9.3	Chris's Reflection	13
10	Weekly blog posts	13
10.1	Fall Week 3: Gabe	13
10.2	Fall Week 3: Luay	13
10.3	Fall Week 3: Chris	13
10.4	Fall Week 4: Gabe	13
10.5	Fall Week 4: Luay	14
10.6	Fall Week 4: Chris	14
10.7	Fall Week 5: Gabe	14

10.8	Fall Week 5: Luay	14
10.9	Fall Week 5: Chris	14
10.10	Fall Week 6: Gabe	14
10.11	Fall Week 6: Luay	15
10.12	Fall Week 7: Gabe	15
10.13	Fall Week 7: Luay	15
10.14	Fall Week 7: Chris	15
10.15	Fall Week 8: Gabe	15
10.16	Fall Week 8: Luay	15
10.17	Fall Week 8: Chris	16
10.18	Fall Week 9: Gabe	16
10.19	Fall Week 9: Luay	16
10.20	Fall Week 9: Chris	16
10.21	Fall Week 10: Gabe	16
10.22	Fall Week 10: Luay	16
10.23	Fall Week 10: Chris	17
10.24	Winter Week 1: Gabe	17
10.25	Winter Week 1: Luay	17
10.26	Winter Week 1: Chris	17
10.27	Winter Week 2: Gabe	17
10.28	Winter Week 2: Luay	17
10.29	Winter Week 2: Chris	17
10.30	Winter Week 3: Gabe	18
10.31	Winter Week 3: Luay	18
10.32	Winter Week 3: Chris	18
10.33	Winter Week 4: Gabe	18
10.34	Winter Week 4: Luay	18
10.35	Winter Week 4: Chris	18
10.36	Winter Week 5: Gabe	19
10.37	Winter Week 5: Luay	19
10.38	Winter Week 5: Chris	19
10.39	Winter Week 6: Gabe	19
10.40	Winter Week 6: Luay	19
10.41	Winter Week 6: Chris	19
10.42	Winter Week 7: Gabe	20
10.43	Winter Week 7: Luay	20
10.44	Winter Week 7: Chris	20
10.45	Winter Week 8: Gabe	20
10.46	Winter Week 8: Luay	20
10.47	Winter Week 8: Chris	20

10.48	Winter Week 9: Gabe	20
10.49	Winter Week 9: Luay	21
10.50	Winter Week 9: Chris	21
10.51	Winter Week 10: Gabe	21
10.52	Winter Week 10: Luay	21
10.53	Winter Week 10: Chris	21
10.54	Spring Week 1: Gabe	21
10.55	Spring Week 1: Luay	22
10.56	Spring Week 1: Chris	22
10.57	Spring Week 2: Gabe	22
10.58	Spring Week 2: Luay	22
10.59	Spring Week 2: Chris	22
10.60	Spring Week 3: Gabe	22
10.61	Spring Week 3: Luay	22
10.62	Spring Week 3: Chris	23
10.63	Spring Week 4: Gabe	23
10.64	Spring Week 4: Luay	23
10.65	Spring Week 4: Chris	23
10.66	Spring Week 5: Gabe	23
10.67	Spring Week 5: Luay	24
10.68	Spring Week 5: Chris	24
10.69	Spring Week 6: Gabe	24
10.70	Spring Week 6: Luay	24
10.71	Spring Week 6: Chris	24
10.72	Spring Week 7: Gabe	25
10.73	Spring Week 7: Luay	25
10.74	Spring Week 7: Chris	25
11	Appendix 1	26
11.1	Python Script Plays the Game	26
11.2	Nodejs Server	27
12	Appendix 2	28
12.1	Detection Layer Result Sample 1	28
12.2	Detection Layer Result Sample 2	29
12.3	Training plot Loss v.s Accuracy	29
12.4	Final Project Setup	30
13	Appendix 3	30
13.1	Benchmark	31

1 INTRODUCTION

Our project was requested by NVIDIA's Deep Learning Institute in order to provide a new training course in neural network training and development for new developers. The goal was to provide an exciting project for beginners that would help ease them in to the growing field of Deep Learning. Deep Learning itself is a field focused on the training and development of Deep Neural networks to be applied to machine-learning based challenges. Deep Learning is a major avenue toward building technology like self-driving cars and automatic facial recognition. Getting more developers involved will support the field's continued growth and lead to important breakthroughs in AI technology.

The project itself was given in an open-ended fashion. We, as a team, were allowed to choose a project that met a few simple requirements. The first was that we had to use a Neural Network to accomplish the task. The second was that it had to run inference on the Jetson TX1 developer kit. With these guidelines set, we decided to use the project to train a neural network to play the video game Galaga, an arcade shooter.

Our client was Mark Ebersole and our team consisted of Gabriel Morey, Luay Alshaw, and Chris Johnson. Mr. Ebersole acted as a supervisor and provided us necessary resources from his company, such as the Jetson TX1 developer kit. He also connected us with people at NVIDIA who could offer advice on technical implementation of the project. Chris was assigned as team captain and provided work building training images, setting up and acquiring hardware, and creating and organizing documentation. Luay was the primary authority on the neural network itself and was in charge of training it and improving its functionality for our task. Gabriel worked on training images, device communication, neural network functionality improvements, deployment testing, and documentation improvements.

2 ORIGINAL DOCUMENTATION

The following is a collection of documents written over the course of the project in close to their original form. They are here to give perspective on the progress made on this project throughout the year. The documents included are the Requirements Document, the Design Document, and the Technology Review.

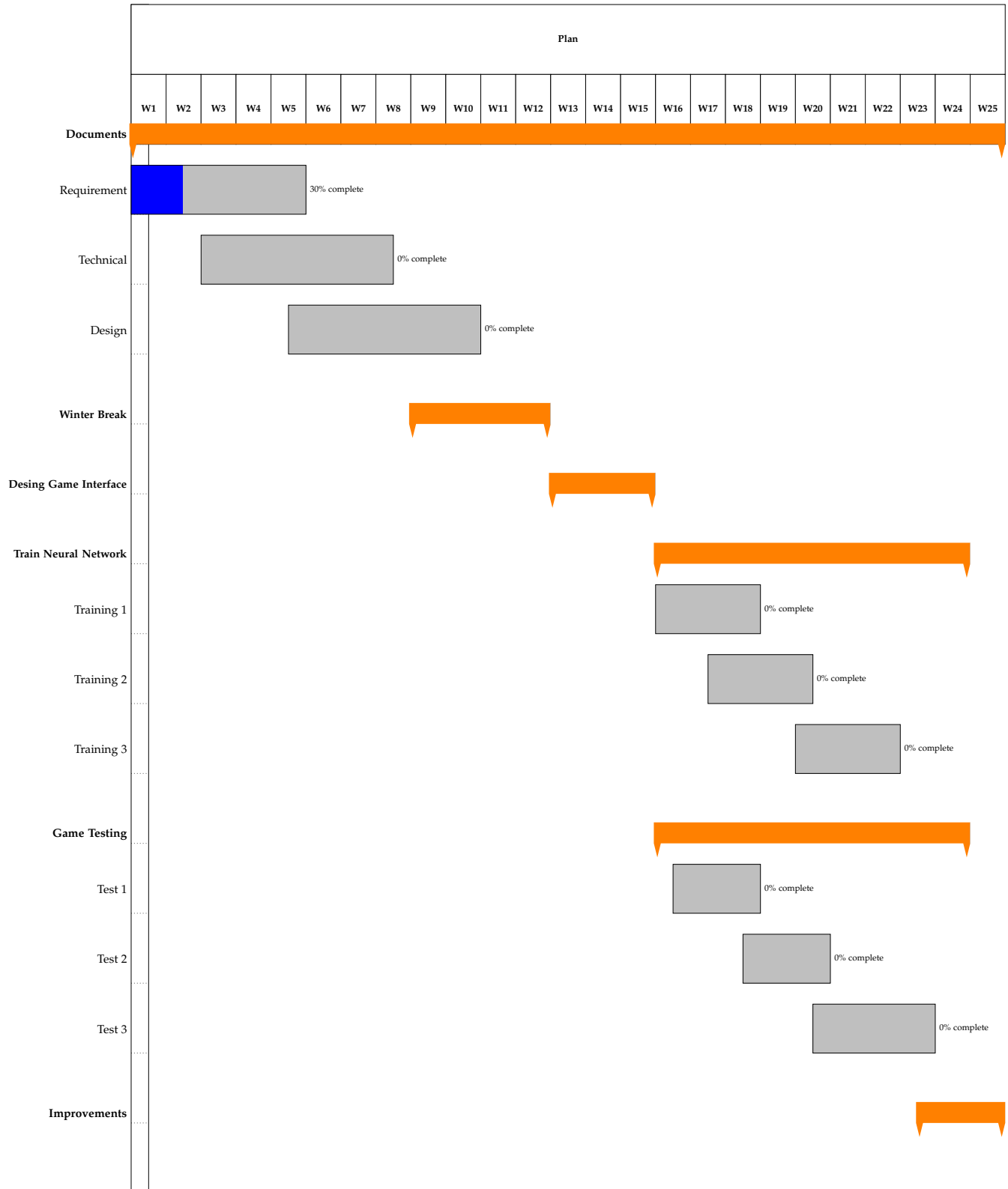
3 CHANGES MADE FROM THE REQUIREMENTS DOCUMENT

3.1 Change Table

1	Requirement	What happened to it?	Comments
2	The neural net must be trained and allowed to create its own code to accomplish the task.	The neural network was trained and does create some of its own code in the form of trained prototext, but it does not create code for the entirety of the problem.	While this requirement changed very little at face value, it is heavily tied to another requirement and thus there have been minor changes to the strictness of its wording.
3	The neural network should define for itself how to respond to in-game stimuli and react accordingly, as per the training it has received.	This requirement was dropped in its entirety. We replaced this with allowing for a custom built decision layer to be set-up on top of the neural network with responses to network output that we as a group defined.	We found that setting up our own neural network structure on top of the one we had already started with would take more development time than we had due to the complexity of developing full network learning layers.
4	The final project must be able to run inference on the Jetson TX1 Developer platform.	This requirement was never changed and was met in its entirety.	There was no unusual difficulty to this requirement and it could not change due to it being part of the core requests of our client.
5	The project must use a high-end GPU cloud to train.	This was more of a design statement and was thus dropped entirely as an actual requirement.	This also ended up being an early and unnecessary design statement as our final neural network did not need cloud computing for its training.

6	We must clearly outline our actions taken and resources required at every step of the process.	This requirement remained unchanged throughout the project.	This requirement was vital to the project, and also easy to meet given the large amount of documentation we had to do anyway. The documentation we've made will be necessary to allow our project to be recreated for training purposes.
7	The solution of our project must be able to beat the first level of the game at least 90% of the time.	This requirement was unchanged.	We initially had trouble meeting this goal, and considered changing it to an easier metric, but within a few weeks of expo we managed to start meeting it and thus made no change.
8	Ideally the system should be able to make it past level 5 approximately 50% of the time.	This requirement was mostly dropped.	This was intended as a possibly reachable stretch goal, but we never achieved this level of performance, so it was not relevant to the final project.

3.2 Altered Gantt Chart



4 CHANGES MADE FROM THE DESIGN DOCUMENT

Several things outlined in the design document's original form were far off base from the reality of the finished product, as is to be expected. When the original design document was written, our team had not fully researched the topic and

we were still learning how to go about accomplishing the task. The first major change from initial design was the choice of neural network architecture. Originally we had planned to use a model based on GoogLeNet to accomplish the task. As our research expanded outward, we discovered SSD, which allowed us to not only detect the presence of objects in a screenshot, but also to identify where they were. We also ended up changing the structure of the decision output. While initially we had intended to build a set of custom learning logic so the neural network could learn to make the game decisions on its own. Due to the difficulty of this relative to the time we had, we scaled back to one layer of in-built decision logic.

When it came to our training methods we also had to re-examine our initial approach. We set up our images in largely the same way as described, but not with any doctoring through OpenCV. We did still do stage one of training, using labelled images to train the network to recognize specific objects. Due to the above changes, training stage two was dropped entirely. There was no longer a component of our project to train as described in stage two. We also did not use any kind of cloud computing for the training, as our own computers proved sufficient.

We never implemented any kind of automatic data visualization for output of the neural network. We did still use a client server system to allow the neural network to communicate with the computer hosting the game. The client-server code was ultimately coded in Node.js. The final hardware layout was pretty much exactly as described, though it could be adapted to use wireless communication and the inbuilt camera on the Jetson TX1.

5 CHANGES MADE FROM THE TECHNOLOGY REVIEW

A number of the technologies discussed in the tech review ended up being superfluous due to some confusion on what the tech review needed to cover. Outside of that, there were several changes that had to be made as we found better solutions. From the first section, we did end up using OpenCV as our primary vision library, though its actual uses in our project were minimal and largely only involved in getting the camera feed through script. The development OS was mainly Linux, as stated, because of the Linux running on the TX1. The discussion of programming languages wasn't necessary, and we ended up using a mix of c++, cuda, python, and node.js. We did use Caffe as our neural network framework, but the neural network architecture we eventually settled on was actually SSD, or Single Shot Multi-box Detector. We didn't use a cloud for training at all, so the cloud section is entirely irrelevant. There was just no need for cloud training when our existing systems could handle the job. The discussions of team communication platform and data visualization also ended up being irrelevant to how we completed the final project. The camera we ultimately used was a small, \$30 Logitech camera. We also used the Jetson TX1's build in camera on occasion. In total, the actual main technologies used were the Jetson TX1, SSD, a Logitech camera, our own computers, a basic computer monitor (multiple models were used for different tests), and ethernet cables.

6 FINAL DESIGN

6.1 Context

This system is not designed for an average computer user. This system is a showcase of possibilities when working with Neural Networks. As such, no real UI will be developed for the purposes of this project. Average people may see the result of the final project by viewing the network play Galaga.

This system is designed for developers. Developers should be able to do a project like this as an introduction to deep learning neural networks. The documentation provided here reflects this focus and is written with developers in mind.

6.2 Summary

The neural network created for this project is able to play Galaga. Its design is based on SSD, a convolutional neural network designed in coordination with Google Inc., University of North Carolina Chapel Hill, and University of Michigan, Ann-Arbor. For more information on SSD see <https://github.com/weiliu89/caffe/tree/ssd>. This neural network underwent one phase of training. The training phase was done with annotated images. The system runs inference on the Jetson TX1 developer kit, and interfaces between a separate computer via communications scripts written in nodejs. The Jetson is linked to the separate PC via ethernet cable. In total, the system learns how to identify objects in Galaga and then uses that knowledge to make a decision on control inputs.

6.3 Neural Network Setup

The neural network set-up itself is a multi-layered combination of components. At the basic level of design, our neural network follows the architecture known as SSD. SSD is designed, as a convolutional neural network, specifically to process images and categorize objects within the images in real time. The network uses a set of bounding boxes and feature maps created through layers of network code to identify individual components of a frame.

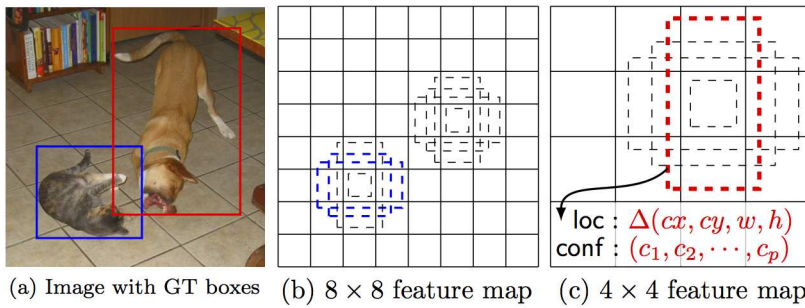


Figure 5.1 SSD Bounding Boxes

The network set-up provided by SSD is built using the deep-learning framework Caffe, so its functions are intimately involved in the final project. Caffe allows for command line instructions to be sent to define the network. To set up the network and Caffe properly we are using the tools provided by Caffe itself and the components found in the SSD branch of the main Caffe repository on Github. In addition to using Caffe resources, the network is coded using a mix of C++ and Cuda, with several scripts in python used to run individual components.

6.4 Communication

In the setup for this project, we have the neural network running inference on the Jetson TX1, but a different computer is hosting the game Galaga. Two elements exist which allow for the network to actually get commands to the game. The first is the decision layer. The decision layer is a hard-coded set of logical responses to in-game stimuli that sits at the very top of the neural network. Coded primarily in Cuda, the decision layer uses the detection output to decide what

command it will send out to the game. The second element is a client-server system coded in Nodejs. The client code is run by the neural network deployment script and sends messages to the server code, which must be running on the machine that hosts Galaga. The server code receives the commands from the system and executes them in-game via key press.

6.5 Hardware Setup

In order to actually use the system, several pieces of hardware are required. The first is an NVIDIA Jetson TX1. The second is a computer capable of playing Galaga and an appropriate monitor. The next is a recording device of some sort. This can be an external camera, the Jetson's in built camera, or a local screen capture device on the Galaga computer. As long as the images are being sent to the neural network on the Jetson, it will work. Finally the system requires an Ethernet cable connecting the host computer and the Jetson or both must be on the same local network. When the neural network is supplied visual feed from the game, the computers are properly linked, and the communication servers are running the network will capably play the game.

7 GUIDE

7.1 Installing Caffe

After cloning the SSD_GALAGA repository you can start installing Caffe. It needs to be installed on the host machine that will be doing the training and on the Jetson TX1. There is a list of dependencies located in `SSD_GALAGA/dependencies.txt` that lists packages based on Ubuntu and some helpful notes. After installing all dependencies you also need to copy `Makefile.config.example` and rename it to `Makefile.config`. Then you need to make changes based on the configuration you have. There are other examples included in the repository that show the configuration file used on the Jetson TX1 and desktop running Ubuntu. Finally, you can run "make -j4" followed by "make py".

7.2 Set up image dataset

Setting up the image set for training and testing requires using annotation methods that can be used to label and identify objects from any given image. In this project, Pascal VOC annotation were used because it's the most common annotation method and is supported by the majority of deep learning frameworks as well as neural networks. The deep learning framework we are currently using supports Pascal VOC and Coco but only Pascal VOC is being used. However, in order to prepare the annotations you need to separate train validation images and test images so you can easily list them in a text file along with their path and XML, which is the annotation file. In addition, you need to write down a label map file that lists all the objects that need to be identified by the neural network. The files are needed in order to create the LMDB database are `trainval.txt`, `test_name_size.txt`, `test.txt`, and `lebelmap_voc.prototxt`. These files located in `SSD_GALAGA/data/VOC0712`. To annotate images we used `labelImage` which is graphical annotation software. It can be downloaded and installed from <https://github.com/tzutalin/labelImg>.

7.3 Training

Before starting the training there is a script needs to be executed to create LMDB database for the training data set. The script is located in `SSD_GALAGA/data/VOC0712/create_data.sh`. The variable `data_root_dir` should be pointing to the data set. After running the script the database will be created and linked to `SSD_GALAGA/examples/`. Some

modifications need to be done on the Python training script in `examples/ssd/ssd_pascal.py`. Also, some values in `solver_param` dictionary which are `base_lr` and `max_iter`. It's recommended to start with 0.01 value for `base_lr`, and then decrease the value till it fits with the data set. Also, the variable `num_classes` needs to be updated based on the total number of objects that will be trained on. The file can be modified if needed. Finally, the script can be run using `"python examples/ssd/ssd_pascal.py"`.

7.4 Play the game

7.4.1 Game Hosting Computer

This computer will run the game as well as the Expressjs server in the background. The server is based on Nodejs, and thus it must be installed. After cloning the keyboardServer repository you need to install the dependencies by running `"npm install"` then `"nodejs ./index.js"` to run the server. The server will receive requests from the Jetson TX1 and will then execute the commands in the form of keyboard press using `robotjs` library.

7.4.2 Jetson TX1

When the training is over you can start using the output model to make it play the game by deploying the model on the Jetson TX1. There is a python script located in `SSD_GALAGA/examples/n_post_play.py`. This script contains a URL variable that points to the IP address of the machine that hosts the game and needs to be updated. Also, `labelmap_file`, `model_def`, and `model_weights` might need to change based on their path. This script captures an image and sends it the neural network to get a decision back. The decision will be sent through the network asynchronously to the other machine that hosts the game. The other machine must be running the Expressjs server listening on port 5000.

8 LEARNING RESOURCES

Over the course of this year we used several online resources in order to research how to use the tools necessary to complete our task. In addition to these online resources we obtained some amount of help and advice from our client and NVIDIA staff he connected us to. Of particular relevance to the project is the SSD repository located at <https://github.com/weiliu89/caffe/tree/ssd> and its accompanying paper available for download at <https://arxiv.org/abs/1512.02325v5>. In addition to the resources for understanding our base neural network we used a lot of articles and guides available through NVIDIA to learn the ropes and figure out how neural networks could be used. NVIDIA's various resources also led us into finding other links and resources. We don't have a full catalogue of the articles we researched, but it was a resource that greatly helped us. We also used video tutorials on their youtube channel such as https://www.youtube.com/watch?v=6eBpjEdgSm0&ab_channel=NVIDIA. Some other helpful papers and tutorials we used include the UFLDL tutorial located at <http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/> and the book *Neural Network Design* by Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, and Orlando De Jesus.

9 LEARNING REFLECTION

9.1 Gabe's Reflection

This project was quite the journey. At the beginning I knew almost nothing about neural networks and how they worked. I was aware of their existence and that they were modelled after the human brain, but I had no idea how complex and

interesting the field was. Over the course of this project I learned about the different kinds of neural networks, their architectures, and their frameworks. I learned how to train and deploy a neural network to accomplish a task. I learned a little bit about custom layer creation too, though not quite enough to build my own network from scratch.

From a non-technical perspective I learned so much about documentation and working in a team environment on major projects. In terms of documentation, my skills were definitely not up to par at the beginning of the term. I honestly think it's still something I need to improve on. However, I have improved greatly because of this project and learned how to express information much more clearly. I also have a much better idea about what kind of information is required on a major project. This has also been an incredible lesson in project management. I've learned how detrimental procrastination is when multiple people are delivering components towards a greater whole. Deadlines are essential to making sure everything comes together, whether they are set by external pressure or a group decision. Group communication is also vital. Everyone needs to know what needs to be done and who's going to work on it.

If I could do it all over again I'd change almost everything. The first thing I'd do is aggressively curb procrastination on the part of myself and others and push for better deadlines. I'd also ensure much better communication and contact with the client, as we had issues with that throughout the year. I'd start researching the project much earlier and I would have spent Winter Break doing more for the project. I would have updated documentation much more often and checked it regularly for necessary changes. Finally, I think we needed a different approach to this project, maybe even a different task more suited to the abilities of the Jetson TX1.

9.2 Luay's Reflection

I have learned many great skills over the year. Although, I have heard about artificial intelligence and machine learning but I never had the chance to interact with them. By working on this project I was able to interact with a class of machine learning which is deep learning. Also, I was able to interact with the Caffe deep learning framework and be able to write customized neural layer using C++. This project is code intensive and I have been using Python, C++, Javascript and shell scripting. Python used as an interface for C++ and I was able to write python scripts that send request to a server, do a screen capture, and webcam capture using OpenCV.

The project works I have learned is mainly gathering client requirements and being able to accomplish the tasks on time. Also, being able to log project progress and what was done in a wiki is something important especially for large projects.

In Project management, I have learned how to divide large tasks to small pieces to accomplish the work. I also tried to use agile methodology to the project by using user stories and iterations. This was accomplished using GitHub. Working with others in a team requires some patience and giving others the chance to express their ideas and thoughts. This was the most important things I have learned.

Going back to Fall term, we had so many struggles putting pieces together when learning deep learning. By that time we did not know what we were doing but we eventually decided to pick Galaga and use the science of deep learning to play the game through video capture of the game. This requires fast detection modal that can work on embedded devices. If I have the knowledge right now in Fall term or winter term I would use unsupervised training to accomplish this task since this type of training works best with games. Also, I would pick another area to work on that solves real life problems such as in medicine.

9.3 Chris's Reflection

I learned a lot while working on this project. The technical aspects would include things such as how to properly document an assignment, how to provide a service for a client, and how to plan for large projects. These skills are going to come in handy with future projects. The less technical knowledge I gathered includes how to talk and interact with a client, how to work with a team and split up responsibilities, and how to find the tools you need for specific tasks. These are skills I haven't had much opportunity to build in the past, so I'm glad I was able to make a lot of improvement on them throughout the year. I've improved my time management and planning skills. I've improved my technical analysis skills. Working with a team specifically I was able to improve my communication and collaboration skills. I feel I improved my ability to share ideas and work with others to try and come up with the best solution. I've learned working with teams requires a lot communication and it's not good to go more than a few days without updating. It's been really fun working on this project that we got so much freedom on. If I could do it all over again I would shift my schedule around during Winter term so I could spend more time on the project instead on other classes. I wished I had done a little more during Winter break when I had so much free time. I would also keep in better contact with our client and try to come up with an official project idea a little earlier. I would keep in better contact with my teammates as well.

10 WEEKLY BLOG POSTS

10.1 Fall Week 3: Gabe

This week was a meeting with our client, idea brainstorming, and the beginning of work on the problem statement. The project is pretty open-ended right now, so we get to make a decision on what to focus on. We've discussed a few ideas with Mark and between ourselves but we haven't hit anything solid yet. We need to figure out to finish the problem statement though.

10.2 Fall Week 3: Luay

On week three we had the chance to talk to our client. We discussed about what we are going to do and Mark helped us with some ideas to think about regarding the senior design project. Also, we started working on the problem statement each person on the team did a section on the problem statement.

10.3 Fall Week 3: Chris

So far we are planning on still brainstorming ideas and figuring out exactly what we want to do for our project. Our client gave us a pretty good description of what we were doing and some examples of projects that used deep learning. Our progress as of this point has involved getting a better understanding of our problem. Our main issue was setting up meetings that worked for everyone's schedule. This is the part I guess where I have to get creative. I'm not sure what to say. I'm excited to get started on the project and see what cool things we can create using deep learning.

10.4 Fall Week 4: Gabe

This week was a bit of a rough week. We had issue meeting with our client and with getting his signature for the problem statement. In fact, we didn't hear from our client at all this week. We've continued to make changes and improvements to our problem statement. It seems like it's almost ready to turn in, but it may require a few more edits. Next week we plan to meet with our client again and hammer out the requirements document. We also plan to get the signature on our problem statement and turn it in. This week has been stressful, but hopefully we can get back on track soon.

10.5 Fall Week 4: Luay

On week four the team met two times to discuss our problem statement and to improve it. Also, we met with our TA Vee. During this week I worked on performance metrics subsection of the problem statement and tried to make an improvement. Also, during our team meeting, we decided what our project is going to be. Our project will use deep learning to identify mesquite and point at them using a laser. The plan for next week is to gather as many requirements regarding the project to make progress.

10.6 Fall Week 4: Chris

We didn't make a whole lot of progress this week. We're still unsure of exactly what we want to do for our project and are waiting to set up a meeting with our client so we can talk about some of the specifics. We plan to discuss ideas with our client to get a better idea of what each one will require.

10.7 Fall Week 5: Gabe

This week started off slow, and we were still having problems reaching our client. However, we finally heard back from them and got our Problem Statement turned in on Wednesday. We then focused on the requirements document, but had issues because we didn't have the opportunity to meet with our client. We made a cursory rough draft with an estimated Gantt Chart that we can discuss with our client and refine. We will hopefully be back on track next week and turn in the final requirements doc on time.

10.8 Fall Week 5: Luay

This week, our team held a meeting to start thinking about the requirements for our project. During the meeting, we decided to do something else using machine learning. The idea is to train a neural network to play the game Galaga by providing a capture of the screen and feed it to the Jetson development board. However, we are struggling with the requirements as we were not able to meet with our client to gather enough requirements for the project. Eventually, we were able to turn in what we have for the requirements and I worked on integrating the chart with latex.

10.9 Fall Week 5: Chris

This week we were able to get the problem statement submitted, turn in a rough draft of our requirements document, and schedule another meeting with our client. There was still a little uncertainty with the project we were going to do and the things that we would need for it but we ended up settling on a game playing project. This project would learn how to play the game galaga by watching game play. After that we were able to get a rough draft for our requirements document. We plan to meet again with our client via skype and hash out some more of the specifics.

10.10 Fall Week 6: Gabe

This week we got our requirements document done and made revisions to our problem statement. It was a very productive, but also hectic week. We ran into a lot of timing issues and couldn't get everything done till Thursday. We didn't hear back from our client when we sent it off, so we had to ask for a deadline extension. Hopefully in the next week we can get everything settled and be back on course.

10.11 Fall Week 6: Luay

This out team had a second meeting with our client. Unfortunately, I was not able to join the meeting due to a problem with Skybe. However, my team members were able to gather enough requirements from our client for the project. During this week I worked on converting what we have written in the requirements document to the IEEE 830-1998. We also met with Kirsten to go over what we had on the requirements document where she helped us on the formatting and gave us feedback. We also went over the problem statements and did the changes based on the feedback we received from Kirsten. Currently, we are waiting for our Client to sign the requirements document.

10.12 Fall Week 7: Gabe

This week we met to revise our requirements document and also turned in our revised problem statement. We got the signature go ahead on the requirements doc, even if it was a little late. Other than that, we divided up responsibility on the tech review. Though we didn't get much work done during the actual week, it should be finished up by the end of the weekend to turn in on time.

10.13 Fall Week 7: Luay

This week the team met again to revise the requirements document. The necessary revisions were made based on the feedback we have received from our client. Also, each member of the team knows what to talk about in the tech review document. My portions of this are to talk about three different technologies and find three alternatives. For technology 1, I will be discussing visual libraries that will be used in our project to understand visual movements from the game, libraries such as open cv, CCV, and simple cv. Technology 2 is going to compare three different programming languages to write the necessary software for the project. Finally, technology 3, is comparing three operating systems in terms of software development environment.

10.14 Fall Week 7: Chris

We didn't get a whole lot done this week. We were able to get a signature back from out client for our requirements document and gave it to our TA who was able to give us some ideas to improve it. It was a pretty busy week. We plan to add improvements to our requirements document and turn it in again before starting on our tech review.

10.15 Fall Week 8: Gabe

This week we finished the Tech Review and tried to start on the design document. The tech review ran into trouble and we couldn't quite finish before it was due. We did send in for an extension and luckily were not the only ones. We managed to turn it in complete by the new cutoff date discussed in class. With that done, we set our sights on the design document and met Friday to get started. We reviewed the IEEE standard and tried to brainstorm a way to approach it, but ultimately didn't get much of the document written. We split off to work on relevant sections by ourselves and reconvene later.

10.16 Fall Week 8: Luay

This week we turned in the requirements documents. Each person in the team talked about three different technologies, each technology compared with three different alternatives. In my portion, I talked about vision libraries where on

my conclusion I decided that we are going to use OpenCV as its the best option. Also, I did a comparison between programming languages that we are going to heavily use and the winning language was python because of the smooth syntax and the good documentation. Finally, a comparison was done to compare three different operating systems that should be used for the development environment and LINUX was the winning solution. Other than the requirement documents, there is also a document design that we started to consider and we have a starting point for the latex.

10.17 Fall Week 8: Chris

Week 8 was another really busy week. We were able to get our requirements document finished and turned in. We've also got a start on our design document and were hoping to be able to send Mark something by Wednesday, just to stay ahead.

10.18 Fall Week 9: Gabe

This week we didn't really get much done. After only about half a week of school, there was the brief holiday break. I certainly didn't accomplish much this week. The next week will be a scramble to finish the design document and make sure we can get it in on time.

10.19 Fall Week 9: Luay

This week was a short week, but I spent time learning open CV as well as Caffe (Deep learning framework). The reason behind learning these tools at this time is that I want to have a general idea of the design process.

10.20 Fall Week 9: Chris

This was a short week because of thanksgiving break. We met up and discussed briefly our design document and tried to hash out roughly what our design was going to look like. I did some research but was not feeling very confident about my understanding of our design. I still have a lot of research to go and am hoping to have made a good dent on the design document by Sunday.

10.21 Fall Week 10: Gabe

This week we finished the design document despite many setbacks. We really buckled down on Tuesday to complete a rough draft, and then met with McGrath on Wednesday and Thursday to revise and improve the document. While the design is still rough and needs improvement, I believe we can make it better as we continue forward. What we have should satisfy our needs for now. We are also starting work on the final progress report and presentation.

10.22 Fall Week 10: Luay

This week I have a good shape of what I am going to do in this project. Also, I finally got myself familiar with the concepts of deep learning, neural networks, training a neural network, and be able to do image manipulation using OpenCV. During this week I gain more confident about our ability to have a fully successful working project that will let the Jetson TX1 be successfully able to play the game Galaga. Although, this week we worked on the design document where each person wrote the parts that are responsible for.

10.23 Fall Week 10: Chris

This week was pretty stressful. We tried to get the design document done as best we could despite still being fuzzy about how the design was supposed to go. We're planning on arranging meeting with people who can better explain details of neural nets so we have a better idea of the best approach. We sent Mark the design document we were able to come up with. We also met up on Friday and spent most of the day working on our project report. We plan on finishing it up a little bit during the weekend and on Monday.

10.24 Winter Week 1: Gabe

This week we met and worked on our Jetson setup. As we discussed the ongoing build for the neural network, we came to the conclusion that more work was needed in order to get it from identifying images to properly playing the game. We looked at some other methods of doing this, but for now it looks like we'll be sticking with Caffe. We've contacted our client for help.

10.25 Winter Week 1: Luay

This week we met at Gabe's place in order to run and test the jetson kit we received from our client. We were hoping to install Caffe and test the Single Shot Detection neural network. We were able to run ubuntu on it. But, then we started to discuss what we currently have and how I did the training using SSD. We realized we were doing something that could be wrong because we are supposed to train the neural network to play the game and not just be able to identify the game objects. So I had to email our Client and ask whether what we are doing is fine. We also about to shift our path away from Caffe and maybe use Neuroevolution of augmenting topologies.

10.26 Winter Week 1: Chris

The first week we had lecture on Tuesday. We met at Gabe's house to discuss the project and start putting it together. We made some progress starting up the jetson and running it with Ubuntu. We also discussed some of the limitations of our project and other methods we could use to improve on it.

10.27 Winter Week 2: Gabe

This week we heard back from our client and got in touch with an engineer from NVIDIA. He answered some of the questions we'd been having and we felt reaffirmed in our initial approach using Caffe. We have decided that sticking with Caffe is the best decision and have decided to implement a custom layer to the neural network to allow it to make in-game decisions.

10.28 Winter Week 2: Luay

This week we decided that we are going to stick with Caffe and maybe we could implement a custom layer on top of SSD that make a decision only. So, by the time we heard back from our client he directed us to an expert and I asked him about our approach which is to implement a custom layer on top of SSD. We got a green light on implementing a custom layer.

10.29 Winter Week 2: Chris

This week our client emailed us back answering some of our question and getting us in touch with a professional. He was able to okay some of our design decisions.

10.30 Winter Week 3: Gabe

This week we had an important meeting to lay out tasks to get done with our plans for the rest of the term. We're now using Githubs issue tracker system to keep tabs on what we're working on. Chris is in charge of getting our hardware setup working right. Luay is in charge of implementing the custom decision layer. I am in charge of setting up a communication protocol for allowing our neural network's commands to be sent to the game, which will be running on a separate computer. We've all taken on the task of capturing and labeling game images to be used in training the neural network.

10.31 Winter Week 3: Luay

This week I started to learn how to write a custom layer using Caffe framework. I feel like I am comfortable working with Caffe and I am hoping to be able to finish it within two weeks from now. So hopefully by the end of week five, we should have a decision layer that can output reasonable results that will be feed to the controller. Also by the of this week, we are having team meeting which we will discuss available tasks and then everyone will be given a task to work on. Tasks are created using the issue tracker in Github to easily keep track of all tasks as well as to know which member is assigned to which task.

10.32 Winter Week 3: Chris

This week we had a team meeting to discuss how the project was looking so far and what we still had to do. We divided up each of the tasks and assigned them. I was in charge of making sure we got all the hardware we needed and that it all worked together.

10.33 Winter Week 4: Gabe

This week in a meeting we decided who would take on the extra duties of team leader. Chris offered to take on the role, and so set about creating the OneNote setup we need moving forward. Luay ran our first major training sprint. I began looking into different ways to set up the communication and found that the emulator had inbuilt functionality for Lua scripting, including its own library of direct control functions.

10.34 Winter Week 4: Luay

This week I run an extensive training on the neural network that took 13 hours to finish. The result was better from the previous training because I also increased the annotated images to a total of 53 for validation and 13 for testing. With increasing the images and their annotations I realized the neural network got better on some of the labels that the precision got increased. As a team, we would need to do a lot of image annotation in order to make the neural network output high precision result. We also met this weekend to discuss who is going to be the leader and set up one note.

10.35 Winter Week 4: Chris

This week we had a lecture and our group met up to discuss where we were at and who would be the team leader. I offered to be the team leader. I have to get oneNote set up and figure out how to share it with the professors.

10.36 Winter Week 5: Gabe

This week we met once more and solidified our plans moving forward to finish the alpha build and paperwork on time. I continued work on my communications program, but switched to python scripts over the lua approach upon being encouraged to create a more generally applicable setup, since the lua script would have been emulator specific. My new project will have to be Windows specific to get the keyboard command access, but some changes can be made to apply it to other operating systems.

10.37 Winter Week 5: Luay

This week I did work on the algorithm which makes decisions based on the detections coming from the detection_out layer. I was able to test it through setting up a screen capture of a second monitor that has the game Galage from the same machine and see the interactions and decisions that are coming out from the neural network. In addition, Chris was able to annotate 100 images of the game which I had to run another training on the neural network. The training took about 12 hours to finish and reached 35k iterations. The network is able to identify objects in the game with higher confidence.

10.38 Winter Week 5: Chris

This week I got into the annotation process of the images so that I could begin making them to improve the testing. It was slow at first but I was eventually able to become decently efficient. I was able to get 100 in two days.

10.39 Winter Week 6: Gabe

This week we made the hustled effort to finish off the midterm report and document edits. I made a great deal of revisions to the requirements document and the tech review in addition to working on the midterm report. I also finished off my communications code, which we made the decision to do as a python server rather than using the lua script method I tried first.

10.40 Winter Week 6: Luay

This week I was able to finish the decision layer implementation which has CPU and GPU support in Caffe. Also, I was able to test it using another machine that has the game Galaga open and receives a request from the Jetson which then a keyboard press event will be processed to simulate player's movement through the game. A server based on node js used to do this job which is installed on a machine that has Galage and the Jetson TX1 sends a post request to the game hosting machine. Obviously, this a temporary solution and we might need to use Gabe's pure implementation of Sockets instead. But, a comparison between these two methods will be tested based on the fastest and efficient way. Finally, we were able to make some modifications on our documents to reflect the new changes were made on the project for the alpha release.

10.41 Winter Week 6: Chris

This week was pretty busy. We had to revise all of our documents to reflect the changes we'd made throughout our process and get our project setup so that we had something that we could demo.

10.42 Winter Week 7: Gabe

This week was a slow week following the midterm report. We showed off the project to our TA to show that the project is working. We're currently getting only about 3-9 frames per second when the game runs though. We're currently looking at ways to improve the frame rate of the game play. I'm working on gathering annotated images for the neural network training.

10.43 Winter Week 7: Luay

This week we did present our project to our TA and show a demo of the Neural network plays the game. Also, this week I worked on the decision algorithm to do some improvements. The original algorithm was mainly developed to make decisions based on defense. That being said the neural network only makes decisions to make the player survive throughout the game. However, the new approach is to stay away from enemies as well as move the player the enemies direction and shoot them. Hopefully, by making decisions that can attack enemies and stay away from them will make the neural network reaches higher levels.

10.44 Winter Week 7: Chris

This was a busy week for most of my other classes so I was able to work a whole lot on the project. We provided a demo for our TA showing that the neural net now moves left and right to avoid bullets and enemy ships.

10.45 Winter Week 8: Gabe

I didn't get much done this week as I was busy with other classes. I did make some annotations for the training, as well as gathered screenshots from the game to annotate.

10.46 Winter Week 8: Luay

This week I worked on improving the training parameters and hoping to improve the detection precision of the game. Unfortunately, the results were the same as before and no improvements. I would assume the reason is that we still need to gather more annotations of the game to improve the output results of the detection layer.

10.47 Winter Week 8: Chris

This was another very busy week for the rest of my classes.

10.48 Winter Week 9: Gabe

This week was very relaxed. I annotated a few more images for the neural network training, but spent most of the time working on other classes. The project is on a very good track with the timetables. Our neural network core is set up and the training process is clear to everyone. We just need to keep feeding more images until the neural network improves its detection. We've also managed to refine resources on the Jetson to get 10 frames per second consistently in the image processing, which allows us better speed of play.

10.49 Winter Week 9: Luay

This week, there was nothing to do on my end. The core of this project is considered complete. Although, training the neural network is still a continues process. Training can be done whenever we have more annotated images of the game. Currently there a total of 160 images for training. We hope to reach over 500 hundred before the EXPO. Annotating an image is a long process and takes time. But, our goal for this term is to be able to make decisions and make the neural network plays the game by sending signal commands to the game host's computer.

10.50 Winter Week 9: Chris

This week I went over some of our documentation and read up a bit more on SSD. We were emailed that we were supposed to have our midterm report signed so I emailed our client for a signature.

10.51 Winter Week 10: Gabe

This week we came together to create a rough draft of the poster. We had some confusion about how exactly to put it together, but I think the end result works, at least as a rough draft. We also planned ahead for what we would cover in our presentation. Since most of the things we talked about at midterm haven't changed, merely slightly improved, it won't be difficult to talk about the project. Moving forward, I'm going to be working on improving the logic of the neural network's decision layer so that it plays the game better and can more often complete levels without dying.

10.52 Winter Week 10: Luay

This week we met to work on the poster, we did our best to fill all the details. Also, for our presentation, we decided to work on it individually, then at the end, we are going to put everything together. In fact, there are no major changes in terms of our work since the presentation we worked on from week 6. Currently, we have a working solution for handling the communication between the Jetson TX1 and the hosting computer for the game. Also, the decision layer has improved since week 6 and now it can fully play the game independently. There is only one problem which is the processing limitation on the Jetson TX1, it can only process one image per 100 milliseconds. This means the Jetson TX1 makes 10 decisions every second. The workaround for this problem is to reduce the frame rate of the game to slow it and hoping there would be enough time for the Jetson TX1 to make decisions in a timely manner.

10.53 Winter Week 10: Chris

This week our group met up to make sure we knew what we had to do for the next two weeks and to get a rough draft of our expo poster ready.

10.54 Spring Week 1: Gabe

This week I settled into my new schedule while thinking about ways to improve the decision layer logic of the neural network. I didn't get a lot of annotations done this week.

10.55 Spring Week 1: Luay

Out of curiosity, this week I had the chance to explore a different type of training in machine learning which is unsupervised training. By using Arcade Learning Environment library which has access to the game data within the ram. Also, by using Deep Q-Network using Caffe on GitHub I was able to test it with ALE and run a training of the ping pong game that took three days. The result was not bad but there is not enough time change direction from what we have done so far.

10.56 Spring Week 1: Chris

This week we had a lecture to prepare us for the coming months. We have only training left to do before expo so this week I started creating annotations for the neural net's image detection.

10.57 Spring Week 2: Gabe

I started working on integrating TensorRT into our neural network this week so that we could improve the lagging fps of the system. While we are now at a much better pace of play than previously, somewhere between 13 and 15 fps, it's still slower than we'd like. Hopefully through the implementation of this optimization method we can improve the performance.

10.58 Spring Week 2: Luay

This week we were able to find a group to discuss our poster with for the extra credit. Also, I started working on improving the defense and attack algorithm. Since Gabe decided to work on trying to run tensorRT on the Jetson TX1 and try to integrate it with SSD. The plan for next week is to have work done on the decision layer algorithm and test it on the Jetson TX1 as well as home computer GPU.

10.59 Spring Week 2: Chris

This week I was able to get some annotations done and we found a group to discuss our poster with. It was a pretty easy week. We're feeling pretty relaxed now that most of the work is done.

10.60 Spring Week 3: Gabe

This week I kept working with researching TensorRT, though it's looking to be difficult to set up. It's possible it might not even be compatible with our current neural net construction, but it's hard to tell as the program is still in testing phases. There's not a lot of documentation out about it right now.

10.61 Spring Week 3: Luay

This week I had given the Jetson TX2 to run and test our project on it. The goal is to compare the performance between both TX1 and TX2. Due to the differences in GPU architecture between both boards and that resulted in running slow GPU performance of Caffe. The solution of this issue could be solved by changing makefile configuration and hopefully, it can be then run faster than TX1. The best result of TX1 gets up to 10 FPS using our custom neural network model. On the other hand, Gabe was working on improving the decision algorithm on Jetson Tx1 and tensorRT. There were some difficulties finding resources on integrating tensorRT. However, I also tried to improve the decision layer as well

as working on the tensorRT. I was able to write a program using the tesnorRT engine but unfortunately, it did not work because tennsorRT does not support custom Caffé layers as this was stated on the release note and this was the complaint error when running the program.

10.62 Spring Week 3: Chris

This week I was able to get more annotations made. We still have yet to meet with our team to discuss our poster. We need to make a final draft soon. We also have to get it to Mark as soon as possible since it will probably take him a while to get it signed.

10.63 Spring Week 4: Gabe

This week we discovered that TensorRT would not work with our custom layer. We have tried signing up for the early test release of the next version, which does support custom layers, but we can't be sure we'll get it in time. Meanwhile, I've been working on the decision layer logic, particularly how the network responds to certain stimuli. I've decided to add more movement when it notices things and to change slightly how it sends commands as a way of ensuring it dodges incoming enemies better.

10.64 Spring Week 4: Luay

Good improvements happen by the end of this week which resulted in making the neural network plays the game and winning stage 1 90% of the time. Also, the highest score reached so far is 15520 which is a record. Also, the neural network was able to reach stage 4 of the game and this is the best result so far.

My plan for this coming week is to install OpenCV 3.2.0 from the source code in order to use the camera on the Jetson TX1 board because the OpcenCv version in apt-get package manager has only 2.4 and this version does not support the Jetson TX1 camera. We are hoping to do a live demo during expo by having one Laptop that has the game and the Jetson TX1. By using the camera on board we are going to save space and be able to do a live demo.

10.65 Spring Week 4: Chris

This week we met up and figured out what we had left to do before the code freeze on May 1st. We realized there was a lot on the requirements document that we still needed to update and weren't sure if we could meet some of them by May 1st. The neural net still isn't performing as well as it should have. We fixed up our poster, got our group photo, and sent the poster to Mark and the professors. We also asked the professors how much we needed to fix with the requirements document.

10.66 Spring Week 5: Gabe

This week we submitted the poster and merged all our github repos for the the final code freeze. Luay has been working on performance. Meanwhile I've been turning my attention to documentation. With Expo coming up and the final code done, we need to get our documentation on track for the end of the term. I am starting with the requirements document, as the scope of the project has changed drastically since we wrote it.

10.67 Spring Week 5: Luay

This week I continue working on installing OpenCV 3.2.0 to use the Jetson TX1 on board camera. The onboard camera is better quality and capable of capturing full HD compare to the current USB camera which is only able to capture up to 720p. The process of installing OpenCV 3.2.0 was painful because there was not enough disk space and had to spend the time to clean unnecessary files. However, after cleaning the disk from unwanted files I was finally able to install OpenCV. Although, the process of installing OpenCV from source code took more than an hour due to the low power nature of the Jetson TX1. By using the onboard camera we might be able to do a live demo during the expo. Finally, we are planning to email our client this week to give him updates on our current progress.

10.68 Spring Week 5: Chris

This week we got through the code freeze after making sure everything we used was in our repo. We also finished our poster and submitted it for printing. Luay recently was able to boost the performance of the neural net using screen capturing instead of video recording. We will most likely continue working up until Expo on May 19th. There are still two more lectures before then.

10.69 Spring Week 6: Gabe

This week we did a lot of important things to prep for expo. The first was meeting with another group and Professor Winters on Tuesday to discuss our pitches and poster. We worked on how we'd talk to people on the expo floor and got a lot of useful insight from the professor and our peers. We also got to work on the midterm report and presentation and divided out duties for how to tackle it. With everything planned out, we will be finishing off the midterm and getting prepped for expo on Friday the 19th.

10.70 Spring Week 6: Luay

This week I was able to install OpenCV 3.2.0 along with OpenCV 2.4. The old version of OpenCV is needed for Caffe, therefore I had to keep both. The progress to make this happen was painful but at the end, I was able to use the onboard camera by using OpenCV 3.2.0 and version 2.4 for Caffe. The plan is to be able to use both the web camera and onboard camera in case we run into reflection issues on the screen. Also, this week we had to meet in person to work on the midterm presentation. During the meeting, we divided everything so that each person knows what to present during the presentation. At the moment I am working on making sure that we have video recorded for the game just in case we run into issues and use it for backup. Hopefully, we never run into issues and be able to do a live demo of the project during EXPO.

10.71 Spring Week 6: Chris

This week we met up to get started on our midterm report and presentation. We set up our project to see how it was doing in its current state and it was able to make it level four and get a new high score of 18230. We got video of this for our presentation and for expo if we decide we can't do a live presentation of our project with the limited table space. We divided up topics to talk about in our video so that we each had an about equal amount to present.

10.72 Spring Week 7: Gabe

This has been quite the project. Were I to start it all over again, I would definitely tell myself to hit research hard sooner. I would also tell myself to spend more time setting things up in winter break. Partially a lot of time was lost due to breaking my leg, but more research would have helped at the very least. I do think things ultimately came together though. I learned a lot of things from working on this project, particularly in the realm of project and time management. Time budgeting is essential to major projects. It's also just as essential to know what to work on at what time and also to be able to revisit and revise plans when things don't work quite right. Of course I also learned how to train and deploy neural networks. I definitely see myself using the management skills, though I don't know if I'll be focusing on neural networks in the future.

The project itself was awesome when it started working. Watching the machine play Galaga was kind of exhilarating, especially after all the work that went into getting there. There were also a lot of fun challenges in getting the network deployed and optimized. What I liked least was how complicated it was to figure things out. I often found myself wishing I was on a project I knew more about beforehand, due to needing to do so much research into Neural Networks to get this going. My teammates were super helpful in the research department, and Luay did an amazing job catching us all up to speed once he made breakthroughs in how the network could work. I think the result of this project is ultimately something our client will be happy with. If we had time to continue this out further, I would want to change our decision layer logic into several more layers designed to train the neural network how to make its own decisions, rather than just being one layer of hard-coded logic.

10.73 Spring Week 7: Luay

This week we were able to present our project during expo and do a live demo showing off our project. I am happy with the accomplishment we have made so far. Going back to Fall term, we had so many struggles putting pieces together for deep learning. By that time we did not know what we were doing but we eventually decided to pick Galaga and use the science of deep learning to play the game through video capture of the game. This requires fast detection modal that can work on embedded devices. If I have the knowledge right now in Fall term or winter term I would use unsupervised training to accomplish this task since this type of training works best with classic games. Also, I would pick another area to work on that solves real life problems such as in medicine.

10.74 Spring Week 7: Chris

Now that expo is over, if I were to do it all over again the thing I would tell myself to do is keep in better contact with the client. He was often busy traveling but If we had contacted our client at least every week we probably could have avoided some contact issues. The biggest skill I've acquired throughout this project would probably planning and documenting a project that a client is requesting and being able to work with a client to deliver what they are expecting. This seems like a major skill and one that I'll be using and improving a lot in the future. I liked that our project was fun and creative. I didn't like that we didn't have enough time do everything we wanted to do on it. While working on the project I learned that my teammates are both smart, creative, and hardworking individuals. If I were the client I would be pretty satisfied with the work me and my teammates have done. The project could be improved if it were learning more than just recognizing images to play the game, and if given more time that's what I would be working on.

11 APPENDIX 1

11.1 Python Script Plays the Game

Listing 1. Python script captures image and feed to neural network to send a decision to the other computer.

```
while True:
    flag, frame = cap.read() # Capture image from the webcam
    start = time.time()
    if not flag:
        continue
    numerate = 0
    image = frame
    transformed_image = transformer.preprocess('data', image)
    net.blobs['data'].data[...] = transformed_image # input the image to neural network
    # Forward pass.
    decision = net.forward()['decision'] # Get decision from the neural network.
    if (decision[0]==0.0): #send right command to move away from enemies
        print "right"
        payload = {'keys': 'x'}
        response = urllib2.urlopen(url, params=json.dumps(payload), headers=headers, callback=callback)
        print response
    elif decision[0]==1.0: #send left command to move away from enemies
        print "left"
        payload = {'keys': 'z'}
        response = urllib2.urlopen(url, params=json.dumps(payload), headers=headers, callback=callback)
        print response
    elif decision[0]==2.0: #send shoot command
        print "shoot"
        payload = {'keys': 'f'}
        response = urllib2.urlopen(url, params=json.dumps(payload), headers=headers, callback=callback)
        print response
    elif decision[0]==10.0: #send right command to find enemies to shoot
        print "right_x"
        payload = {'keys': "_x"}
        response = urllib2.urlopen(url, params=json.dumps(payload), headers=headers)#, callback=callback
        print response
    elif decision[0]==100.0: #send left command to find enemies to shoot
        print "left_z"
        payload = {'keys': "_z"}
        response = urllib2.urlopen(url, params=json.dumps(payload), headers=headers)#, callback=callback
        print response
```

```
cv2.imshow('video', frame)
cv2.waitKey(1)
```

11.2 Nodejs Server

Listing 2. Nodejs script runs a server and listen for client request to execute keyboard commands.

```
var express = require('express')
var app = express()
var robot = require('robotjs');
var robotjs = require('robot-js');
var bodyParser = require('body-parser')
var keyboard = robotjs.Keyboard();

app.use(bodyParser.json())

app.listen(5000, function () {
  console.log('app listening on port 5000!')
})
app.post('/api/sendkeys', function (req, res) {
  var key = req.body.keys;
  res.json({ success: 'message' });
  if (key=="z")
  {
    keyboard.press(robotjs.KEY_Z);
    robotjs.Timer.sleep (160);
    keyboard.click(robotjs.KEY_D);
    keyboard.release(robotjs.KEY_Z);
  }
  else if (key=="_z")
  {

    keyboard.press(robotjs.KEY_Z);
    robotjs.Timer.sleep (25);
    keyboard.click(robotjs.KEY_D);
    keyboard.release(robotjs.KEY_Z);
  }
  else if(key=="x")
  {
    keyboard.press(robotjs.KEY_X);
    robotjs.Timer.sleep (160);
    keyboard.click(robotjs.KEY_D);
```

```

        keyboard.release(robotjs.KEY_X);
    }
    else if(key=="_x")
    {
        keyboard.press(robotjs.KEY_X);
        robotjs.Timer.sleep(25);
        keyboard.click(robotjs.KEY_D);
        keyboard.release(robotjs.KEY_X);
    }
    else if(key=="f")
    {
        keyboard.click(robotjs.KEY_D);
    }
}
})

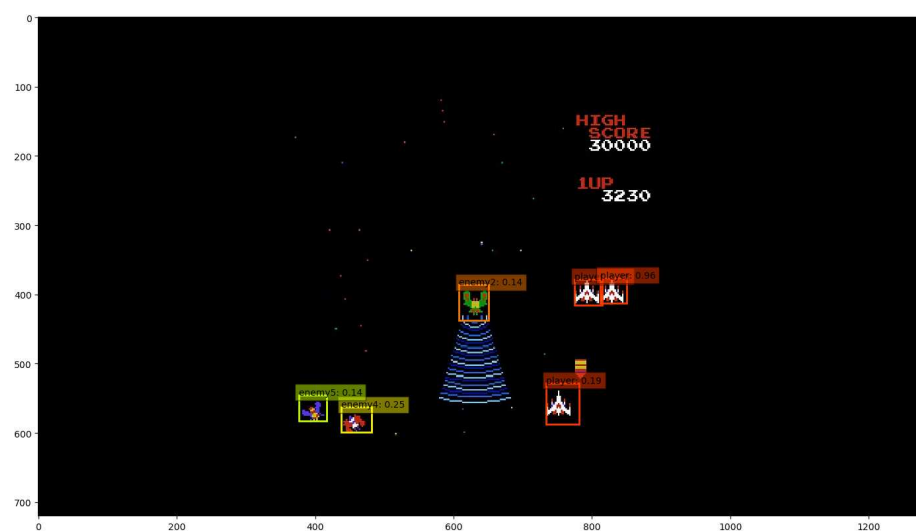
```

12 APPENDIX 2

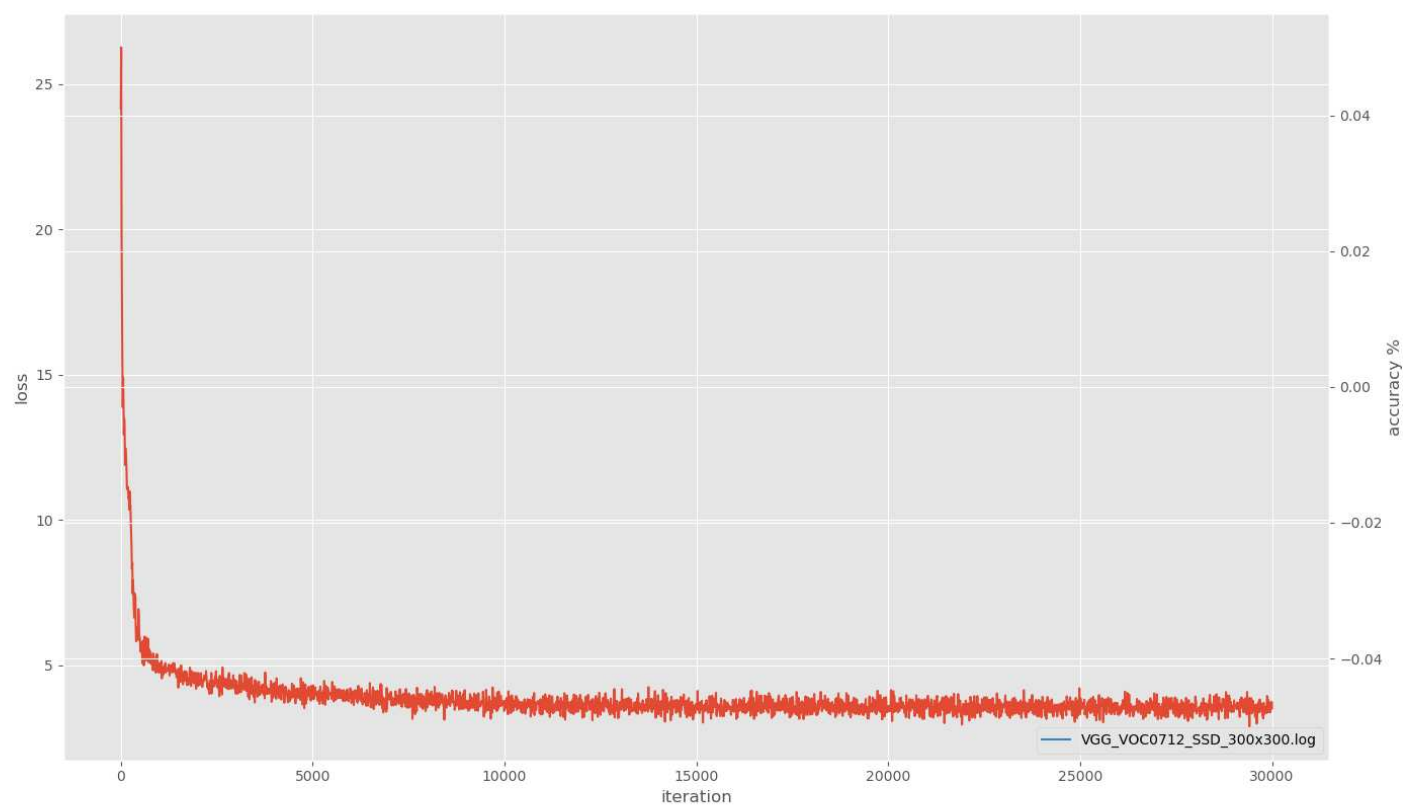
12.1 Detection Layer Result Sample 1



12.2 Detection Layer Result Sample 2



12.3 Training plot Loss v.s Accuracy



12.4 Final Project Setup



13 APPENDIX 3

The benchmarks were obtained by running `python examples/ssd/ssd_pascal_video.py`. The results obtained based on running a video capture of 1000 frames of the game and on three different hardware. The performance of running SSD_GALAGA repository on the Jetson TX2 is tremendously slow. The issue can be related to the new GPU architecture on the Jetson TX2 and could be a configuration problem. Both Jetsons were on their highest performance by running the `jetson_clocks.sh` script before the benchmarks. By the time writing this report, TesnorRT does not support custom layers, therefore inference was based on the weight obtained from Caffe.

13.1 Benchmark

Platform	Layer	Forward pass (1000 iterations)	Average frame per second
i7-6700k 960 GTX GPU	detection_out	0.058727 ms	17 fps
Jetson TX1	detection_out	0.130591 ms	8 fps
Jetson TX2	detection_out	0.270585 ms	3.6 fps