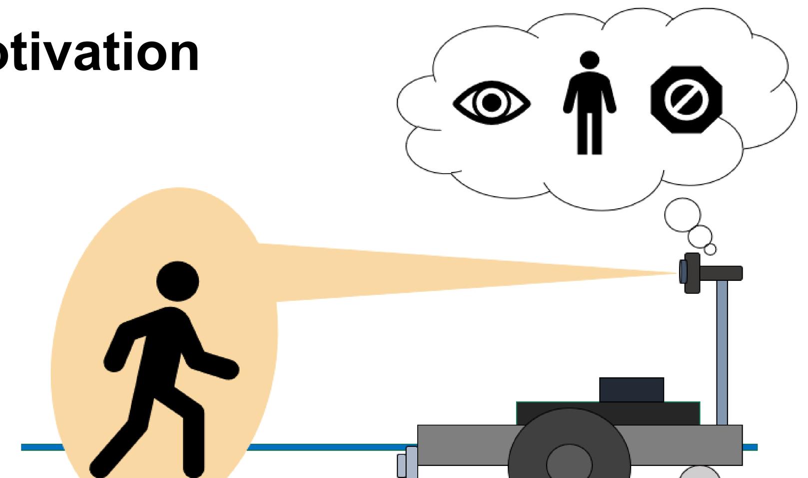
Optimizing a Convolutional Neural Network for Autonomous Jetson Image Classification

Michael Gee[1], Eddy Rodriguez[1], Jeffery Mattos[1], Jose Reyna Pardo[1]
Graduate Mentor: Kevin Yamada[2], Advisor: Hao Jiang[2]
[1]Cañada College 4200 Farm Hill Blvd. Redwood City, CA 94061
[2]San Francisco State University: Dept. of Engineering 1600 Holloway Ave. San Francisco, CA 94132

Motivation



- The end goal of our project is to create a trained and tested model that can be deployed on the Jetson to mimic autonomous self-driving vehicles

1. Introduction

Statement – Build a deployable autonomous robot designed for the Jetson TX1 to enable live image classification.

Goal 1 – Build and train custom dataset

Goal 2 – Deploy and optimize a standard neural network on the Jetson TX1

Goal 3 – Deploy and optimize parameters autonomous field testing

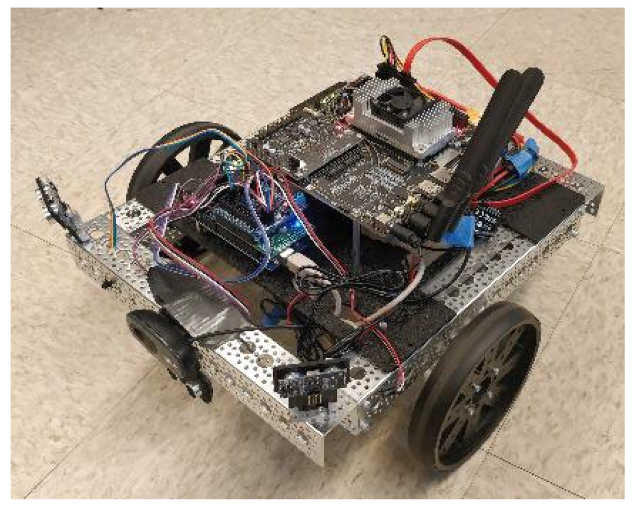
Goal 4 – Enlarge custom dataset with more environmental variation

Goal 5 – Create and compare custom neural networks with a standard neural network for autonomous driving

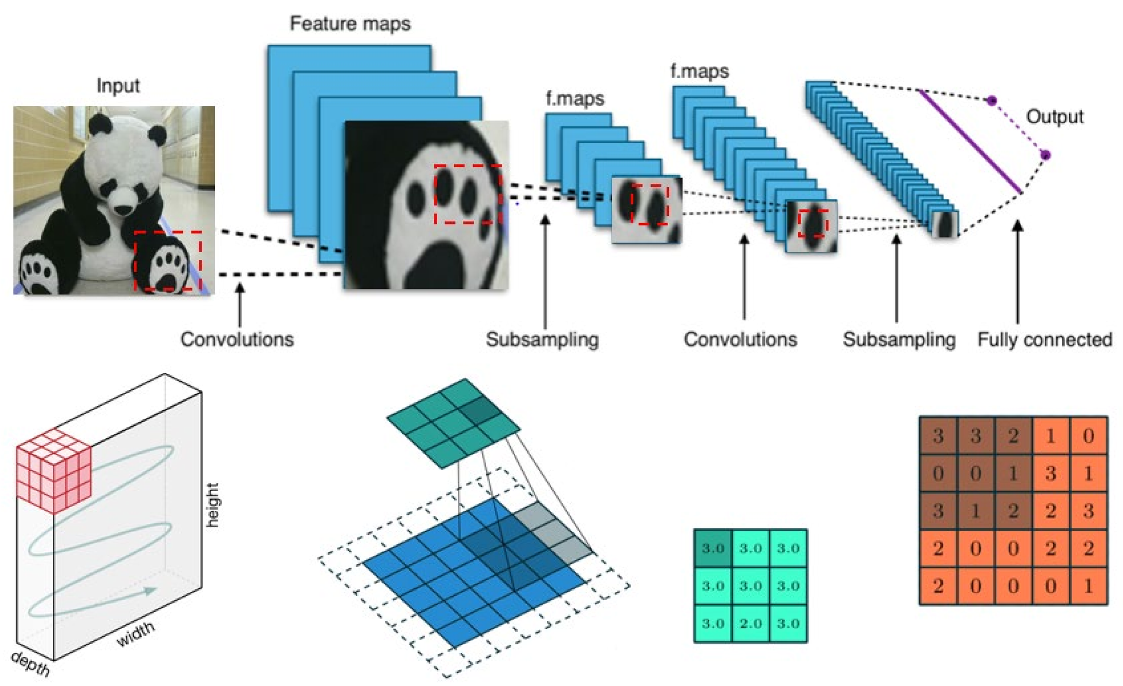
- To teach the Jetson image classification, we'll use deep learning
- Deep learning refers to deep artificial neural networks
- Deep learning is a branch of machine learning and machine learning is a branch of AI
- The word "deep" refers to the number of layers, as you increase the number of layers, neural networks have the capacity to become more intelligent
- With more layers computational training becomes more intensive

2. Platform: Jetson TX1

- Jetson TX1 is a fast, power-efficient embedded AI computing device
- Jetson uses GPU and CPU for parallel computations
- Jetson TX1 allows for edge computing in a small form factor



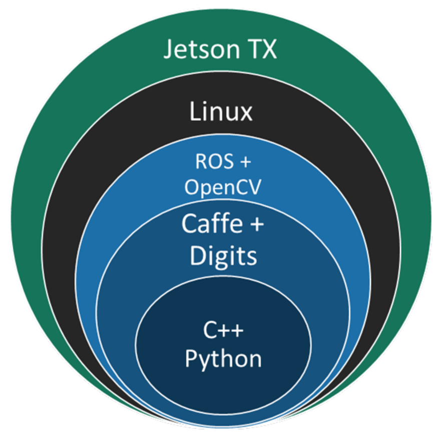
3. Convolutional Neural Network



- A neural network are a set of algorithms modeled after the human brain designed to recognize patterns
- A Convolutional Neural Network is a deep learning algorithm that has two unique abilities:

1. Able to assign significance to various aspects of an object in an image
2. Able to differentiate an object from one another within the same image

4. Software Hierarchy



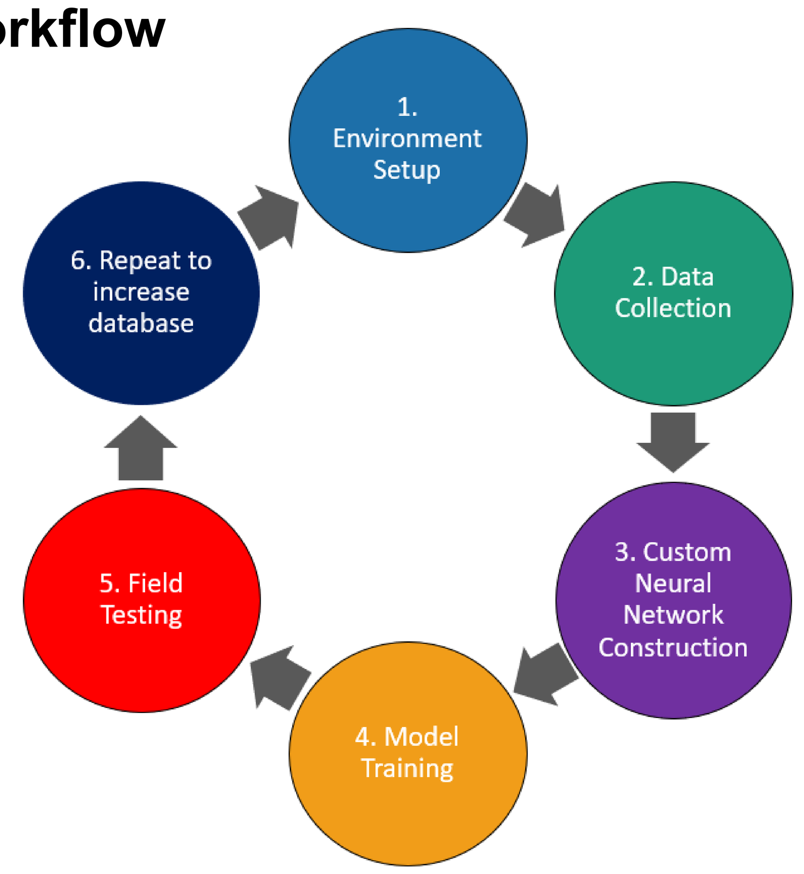
ROS / Robot Operating System
- A framework that simplifies the process of controlling the Jetson.

OpenCV
- Used for real-time computer vision. Process image's pixel-to-pixel manipulation

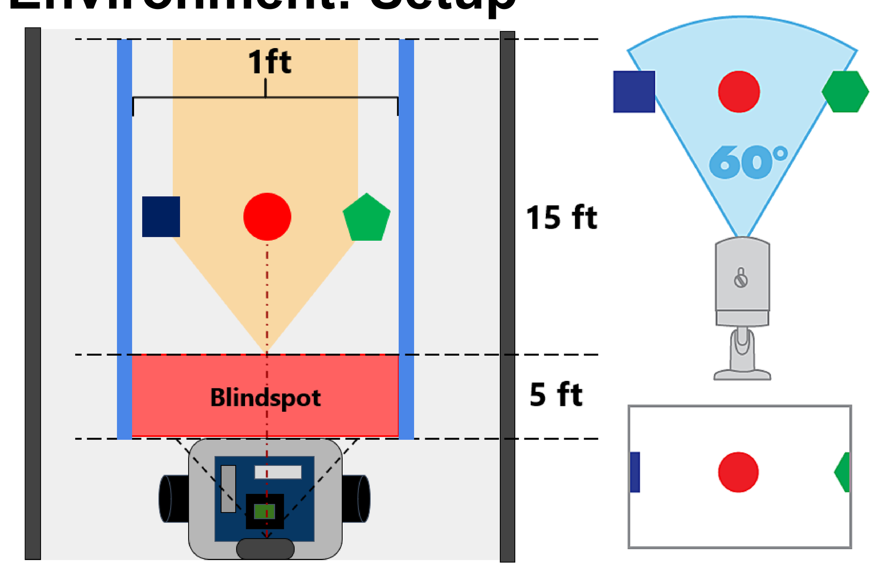
DIGITS
- Provides graphical web interface to Caffe framework instead of dealing with the command-line interface

Caffe
- A framework for deep learning

5. Workflow

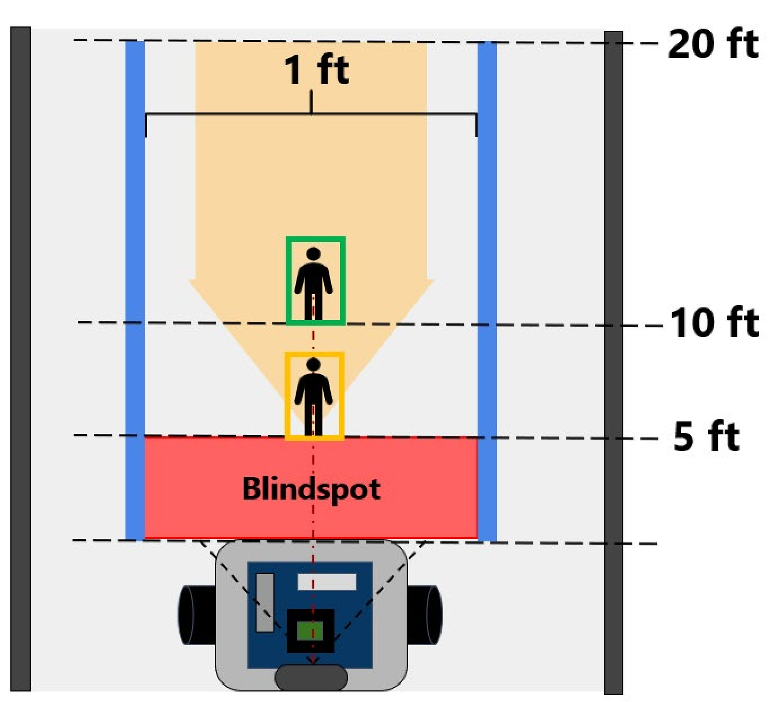


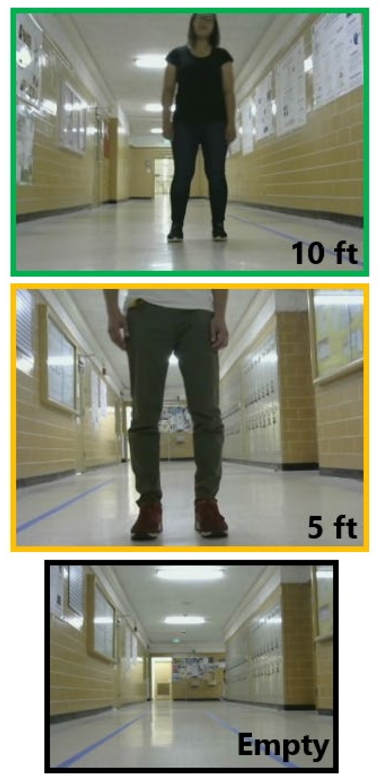
5. Environment: Setup



- Setup located in Hallway
- Road spans 20 ft in length and 3 ft wide
- With a rear viewpoint, there's a 4ft blind spot
- Webcam films at 320 x 240 resolution with 30 FPS

6. Data Collection

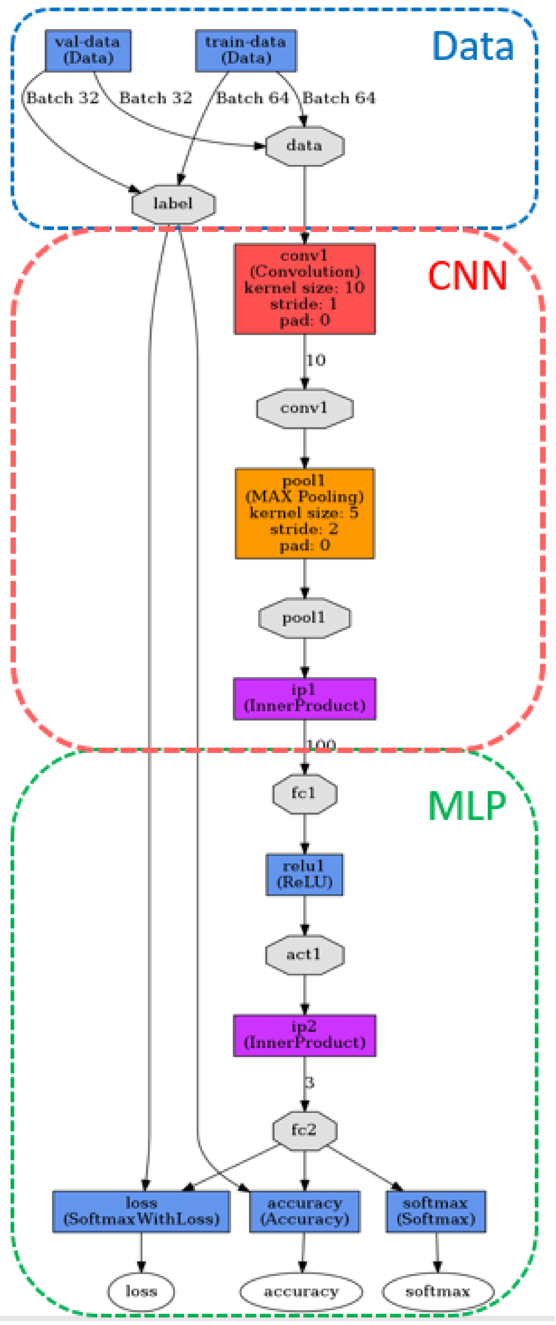




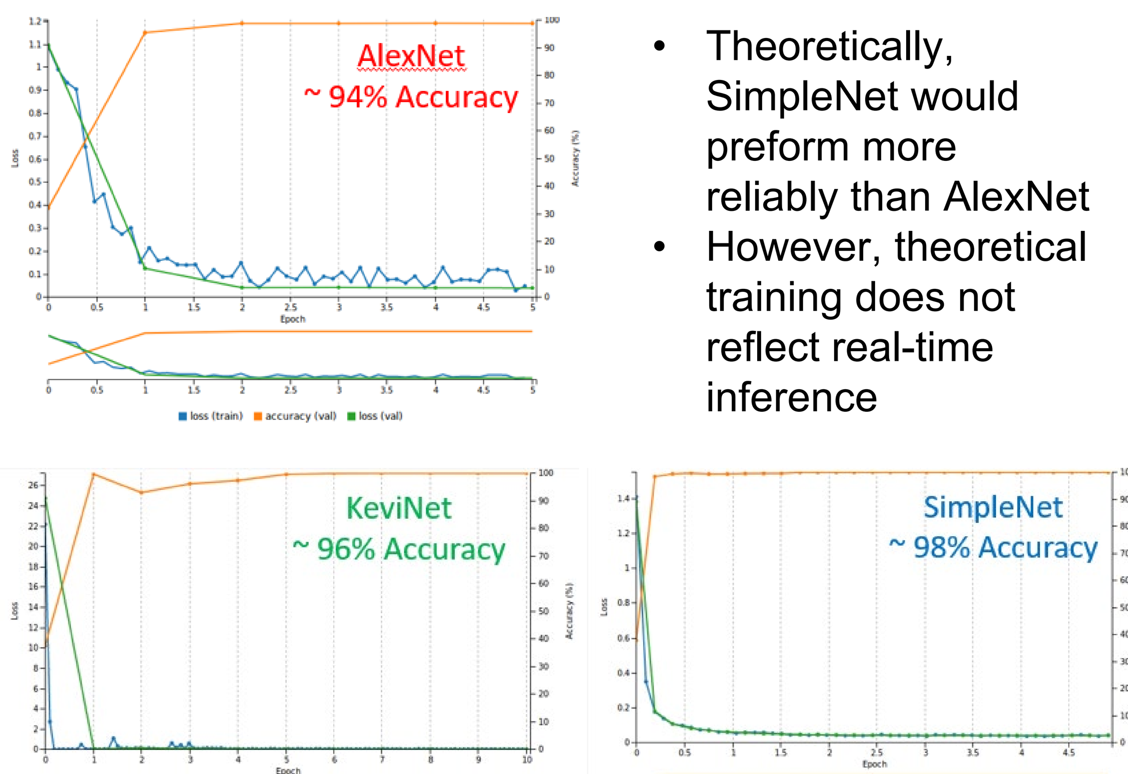
- Data Iterations:
- V1 = 4,091
- V2 = 11,687
- V3 = 34,529
- V4 = 51,221
- V5 = 60,662
- Classes:
- Empty
- 5ft
- 10ft

7. Custom Neural Network

SimpleNet = MLP + CNN
- Convolutional Layer
- Pooling Layer
- Fully Connected Layer = 2
- Activation Function: ReLU
- Softmax
- Loss
- Accuracy



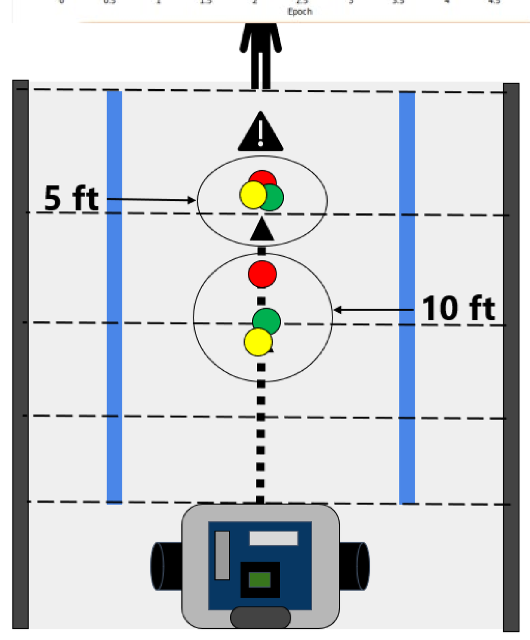
8. Model Training



- Theoretically, SimpleNet would preform more reliably than AlexNet
- However, theoretical training does not reflect real-time inference

9. Field Training Results

- Testing was performed on 4 test subjects for 5 trials
- 2 controlled subjects, part of the trained data
- 2 uncontrolled subjects, not part of trained data



Neural Network	5ft averages	10ft averages	5ft STD	10ft STD
AlexNet	4.694	10.805	0.728	2.053
KeviNet	4.657	9.398	0.753	0.555
SimpleNet	4.212	13.388	1.552	1.063

10. Conclusion

AlexNet	KeviNet	SimpleNet
• Robust CNN • Large overhead • Contains 8 layers • Activation Function – ReLU • 256 x 256 image input • Inference Speed = ~35s • Most Accurate	• Lightweight CNN • Moderate overhead • Contains 2 layers • Activation Function – ReLU • 320 x 240 image input • Inference Speed = ~21s • Fairly Accurate	• Multi-layer Perceptron • Small overhead • Contains 1 layer • Activation Function – Sigmoid • 320 x 240 image input • Inference Speed = ~19s • Least Accurate

11. References

[1] Iandola, F., & Keutzer, K. (2017). Small neural nets are beautiful. *Proceedings of the Twelfth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis Companion - CODES 17*. doi: 10.1145/3125502.3125606

[2] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks fromOverfitting. *Journal of Machine Learning Research 1*.

[3] Jia. Y, Shelhamer. E, Caffe

12. Acknowledgements

Thank you to Dr. Hao Jiang and Kevin Yamada for mentoring us in this project with supportive guidance. Finally, we are truly grateful for this opportunity brought to us and led by Nick Langhoff through the US Department of Education Minority Science and Engineering Improvement Program (MSEIP), "Accelerated STEM Pathways through Internships, Research, Engagement, and Support" (ASPIRES), Grant No. P120A150014.