Contents

List of Figures

List of Tables

Preface

Chapter 1: Introduction to DATA61, CSIRO

This chapter describes the history and main business activities of this company.

Chapter 2: Training Experience

Chapter 3: Conclusion

Acknowledgment

Abarajithan Gnaneswaran,

Undergraduate,

Department of Electronics and Telecommunication Engineering,

University of Moratuwa.

1 Introduction to Training Establishment

1.1 Company Overview

Commonwealth Scientific and Industrial Research Organization (CSIRO) is the Australian federal government agency for scientific research and development. CSIRO has its headquarters in Canberra, Australia and several branches across the world, with over 5500 employees. CSIRO is known for the development of Wi-Fi, Atomic absorption spectrography and the polymer banknote which have changed the lives of millions of people around the world.

CSIRO consists of many parts: Agriculture and Food, Data61, Energy, Land and Water, Mineral Resources...etc with research centers in several cities of Australia. DATA61 is a part of CSIRO that aims on developing a data driven future for Australia. DATA61 consists of multiple groups: robotics and automation group (RAG), data privacy group, mobile security group, distributed sensor networks...etc.

I worked in the Pullenvale (Brisbane) branch of CSIRO. It is called the 'Robotics hub of Australia' due to the large number of robotics projects, facilities and researchers present in the Pullenvale branch. The robotics and automation group of CSIRO is known worldwide for their state-of-the-art SLAM (Simultaneous Locomotion and Mapping) algorithms.

Fig. 1.1. DATA61 logo

1.2 Company History

It was formed in 2015 by merging NICTA (National Information and Communications Technology Australia Ltd) with CSIRO's data science section.

1.3 Organization Structure and Hierarchy

1.4 Areas of Interest

1.5 Current Situation

The RAG of CSIRO was recently selected as one of the six teams worldwide for the DARPA Subterranean Challenge by United States Department of Defense. Therefore the next four years of research in Robotics in CSIRO will be more focused on developing robots that can simultaneously map and navigate underground tunnels, caves and mines without GPS or reliable communication with humans. For this task, incorporating machine learning into the workflow of algorithm development and testing is of paramount importance for all researches in RAG. I addressed this problem by developing an efficient end-to-end pipeline for this and demonstrating it through two projects.

1.6 Impacts on Sri Lankan Industry

1.7 SWOT Analysis

1.7.1 Strengths

1.7.2 Weaknesses

1.7.3 Opportunities

1.7.4 Threats

1.8 DARPA Subterranean Challenge

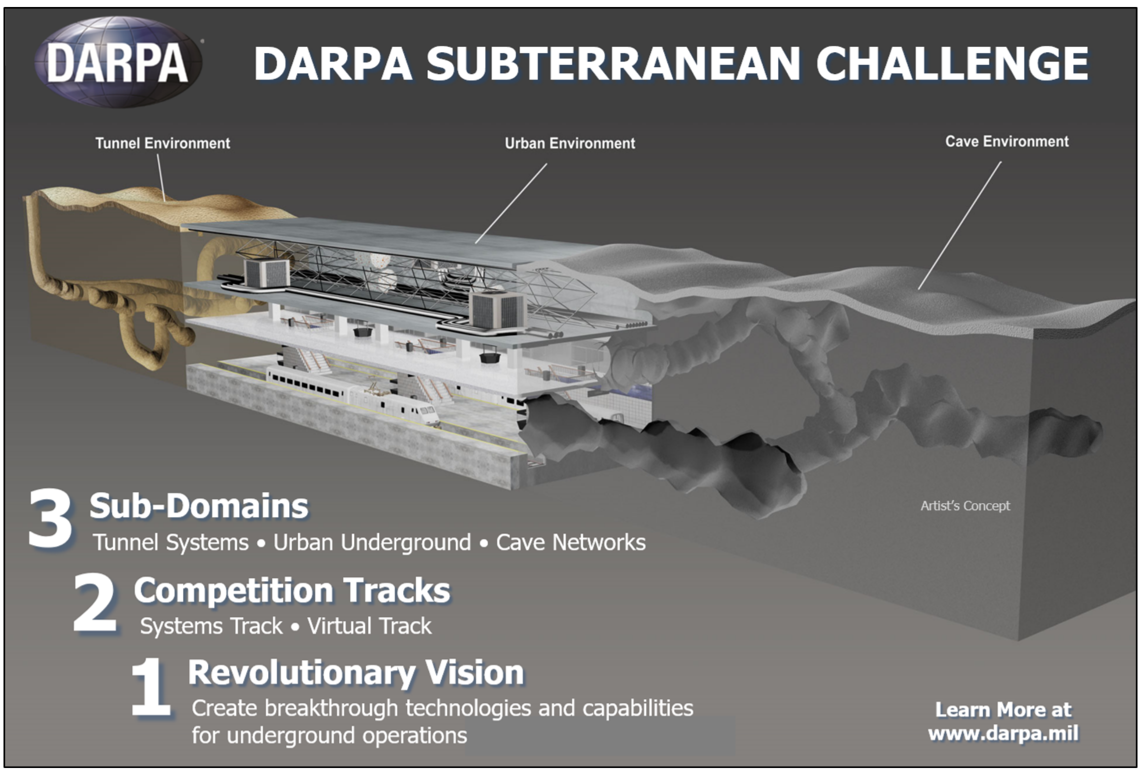

Fig. 1.2. CSIRO focuses on DARPA challenge

1.9 Usefulness to the Country

1.10 Suggestions to Improve the Company

2 Training Experience

2.1 How I got the Opportunity

After graduation, I wanted to continue doing higher studies and become an academic, rather than settling for a job at a company. Therefore, for my internship, I applied for research opportunities in universities and institutes around the world. I got positive response from two or three institutes, one of them being CSIRO. I sent my CV to Dr. Navinda Kottege from RAG, DATA61, CSIRO in February 2018, requesting a research internship opportunity. He asked me to complete a set of 3 timed tasks online to assess my skills in programming and algorithms. He then interviewed and offered me the position as research intern student in CSIRO for 6 months.

Initially I was informed that I am being assigned to the project titled "Computer vision based off-board autonomous UAV Navigation" under the supervision of Mr. Frederick Pauling, a highly capable and friendly senior engineer in DATA61. I was informed that knowledge in ROS (Robot Operating System) and Tensorflow would be necessary, so I spent few weeks learning the basics before the internship.

However, when I arrived at CSIRO, Mr. Frederick Pauling had been promoted into the Group Leader of RAG (Robotics and Autonomous systems Group), to lead the cutting edge robotics research in Australia. Therefore, I could not be assigned into the said project under his supervision. As a result, Uvindu and I was assigned under the supervision of Mr. Nicolas Hudson.

Mr. Nicolas Hudson arrived CSIRO only few weeks before us, after working as a senior roboticist in NASA's Jet Propulsion Laboratory, Boston Dynamics and Google's Machine Learning Division. In CSIRO he wanted to streamline the workflow of the RAG group and incorporate Machine Learning tools into their workflow seamlessly. He asked us to work with him in one of his experimental projects: "Learning Transfer Across RGB, Thermal and IR Modalities in CNNs". After about a week, Uvindu requested for a project that is more focused on hardware. Hence, he asked us to modify Trailnet for autonomous indoor navigation.

2.2 Trailnet: A Classification Network for Autonomous Trail Navigation

2.2.1 Trailnet: An Introduction

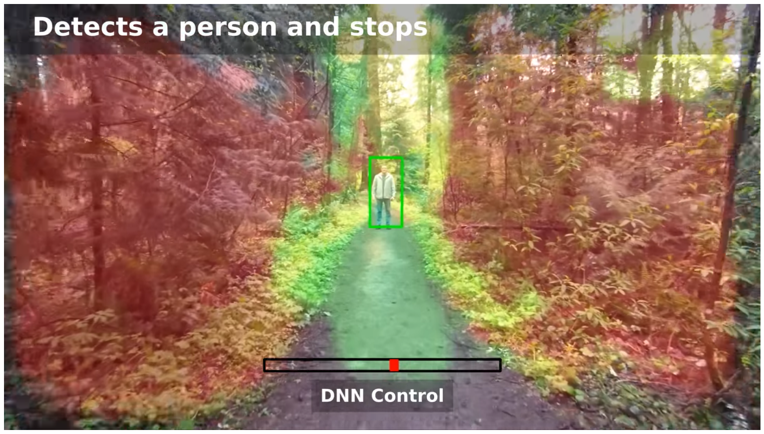

Fig. 2.1. NVIDIA's Trailnet Navigating a Drone

In 2017, four researchers published a paper titled "Toward Low-Flying Autonomous MAV Trail Navigation using Deep Neural Networks for Environmental Awareness" [3] (Trailnet paper in short) with the funding of NVIDIA. The paper describes the following:

- Merits of approaching autonomous navigation as a classification problem
- Architecture of their Trailnet CNN, a modified version of Resnet-18. [2]
- Data collection techniques for the Trailnet CNN
- Training Trailnet with a relatively small dataset
- Hardware hierarchy and the command flow between cameras, NVIDIA Jetson TX2 [1] running ROS and the flight control board.
- Usage of YOLO for obstacle detection and algorithm for obstacle avoidance.
- Results and observations after flying the quadcopter autonomously in the forest trail for several kilometers.

2.2.2 Navigation as Classification

Their model was based on a concept discussed in a 2016 paper : approaching autonomous navigation as a classification problem. That is, given an RGB image the CNN would output three probabilities: that of the camera facing left, right or center with respect to the trail. The key advantages in this approach are:

- The effect of noise introduced by human errors during data collection on training are minimized due to discretization.
- Data collection and labeling is straightforward
- Performance can be fine tuned, by adjusting K1 and K2 accordingly. See Figure: 2.2
- The depth of the required neural network is less, compared to the regression network that provides similar accurate performance.
- Can train the relatively shallow network with a relatively small dataset and shorter training time.
- Can be implemented on low powered devices.

In the Trailnet paper They choose Resnet 18 as the basis for their architecture since it is small enough to be run on real time in a power-limited device like Jetson TX2. Resnets (Residual Networks) are special kinds of deep neural network that uses "short circuits" between layer outputs to prevent the problem of vanishing gradients, as a network gets too deep. By employing this technique, researchers have been able to create networks that are thousands of layers deep and still outperform shallower networks. Resnet-18, Resnet-50...etc are popular variants of applying this technique on deep convolutional neural networks.

Trailnet is not an RCNN. That is it does not remember past inputs nor correlate current inputs with past and future values for prediction. It is a simple CNN that gives a twist command based on the current image. Input to trailnet is a 320 x 180 x 3 RGB image and the outputs are six softmax nodes connected to the output of a slightly modified resnet-18. The six output layers signify the probability of the given image facing left, center and right and the robot (or UAV) being aligned left, center and right on the path. Weighted (by adjustable constants k1, k2) sum of these probabilities provide the angular twist command, which is used to steer the robot. This additional consideration of alignment, prevents the UAV slowly drifting off the center of the trail and crashing with tree branches near the trail edges. Together, the facing and align probabilities correct the course of the UAV to stay in the center of the path. YOLO and SLAM are used for obstacle avoidance.

Data collection was done using a camera rig with 3 cameras facing at three angles (left, center and right). The rig was carried by a person along a forest trail. The video feed from each camera had been then labeled accordingly. Similarly align to left, center and right data has been collected. A pretrained network (previously trained on IDSIA dataset of 40,000 images) had been fined tuned with this collected data. After training, Trailnet was run with ROS (Robot Operating System) and Caffe on Jetson TX2 on board the UAV. Jetson TX2 receives the video frame from camera, processes it and sends it sends command to the flight controller.

2.2.3 Building Trailnet from scratch

One objective of their project was to showcase the capability of NVIDIA's Jetson TX2 high level controller board. Therefore, they had used Caffe framework to build the network and DIGITS framework to train it. However, our key objective in working with this was to build a unified pipeline which all scientists in DATA61 can use. Since most of them were familiar with TensorFlow and since it is the state of the art framework today, I rebuilt the 20-layer network in TensorFlow-Keras and trained it from scratch in CSIRO's supercomputer as I built the pipeline.

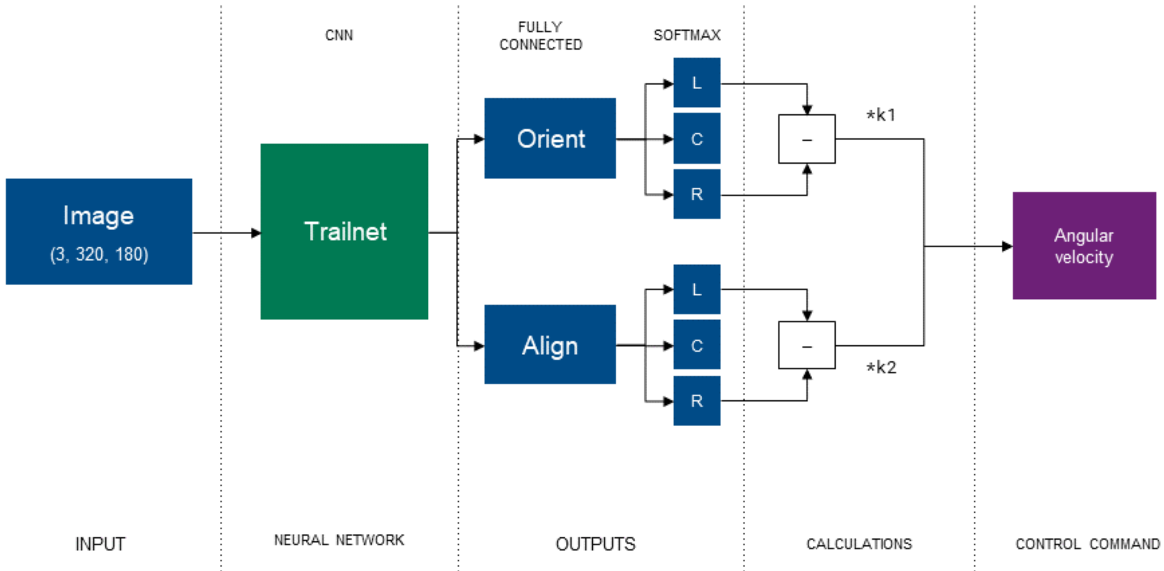

Fig. 2.2. Simplified Trailnet Architecture and Post Processing

2.2.4 Data collection and training Trailnet from scratch

Our goal with this project was to train a robot navigate indoor hallways as a demonstration of our end-to-end pipeline. Hence we mounted three cameras on the robot, facing center, left and right by 30 degrees. We took the robot along the hallways of CSIRO using a remote control and recorded the image stream data as ROS bags. In each hallway, we took the robot on three paths: center aligned, left aligned and right aligned. We then extracted the image stream into an image dataset by taking one image every second (1 fps) from the image stream. The resulting hallway dataset consisted of 120,000 images.

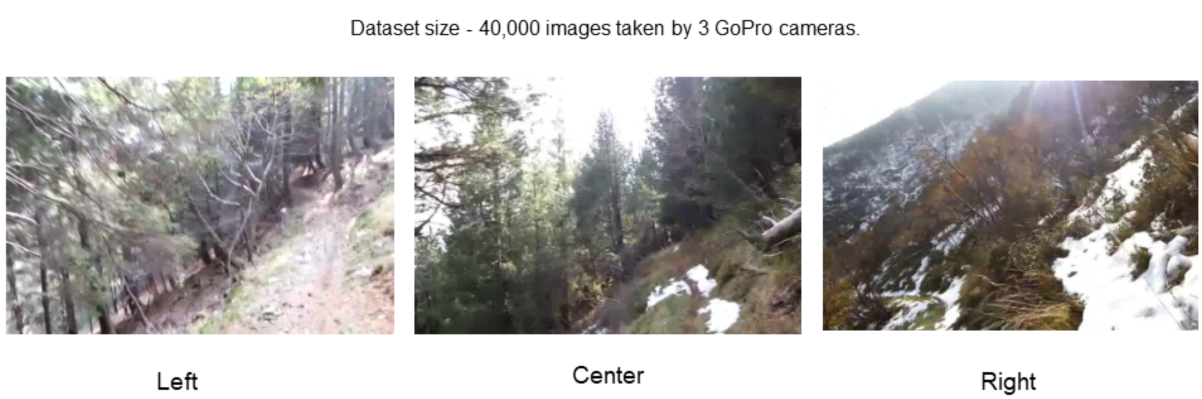

Fig. 2.3. IDSIA Dataset of 40,000 images

Fig. 2.4. Hallway Dataset of 120,000 images

The 120,000 images were stored in the supercomputer and used to train the network. First, align output nodes of Trailnet were frozen and the network (with facing output nodes only) was trained on the IDSIA dataset of 40,000 images. Then, the same configuration was trained on the hallway facing dataset. Finally, the facing-output nodes were frozen and the rest (align-output nodes) were trained on the hallway-align dataset.

Training the network was a laborious process prone to errors. The supercomputer sessions automatically terminate every few hours, requiring me to stay by the CSIRO provided desktop throughout the process. I stayed overnight for five days alone in the office to train this and the other networks.

2.2.5 Deployment on a Robot and Testing

The trained model was optimized into a tf-trt graph (See section: 2.4.3) and executed inside a python based ROS node. The latency was 20 ms, which was enough to process an input image stream at 30 fps during inference. My ROS node also performs necessary calculations and publishes a velocity message (type: geometry_msgs/Twist) to a topic that is subscribed by the motor controller and the predictions (type: Float32MultiArray) for debugging. I also designed it in a way that the constants K1, K2 can be tuned by publishing the constants to a topic that is subscribed by the Trailnet ROS node.

After setting up this way, the robot was tested for its ability to navigate the hallways autonomously. It's response to external disturbances was checked by kicking the robot in either direction, moving it off the center of the track. K1, K2 constants were tuned to provide the shortest response time while maintaining a steady speed when undisturbed. We also created a visualization technique, where the predictions and commands of the robot can be visualized with the image stream. Following images show the testing process and the corresponding visualization.

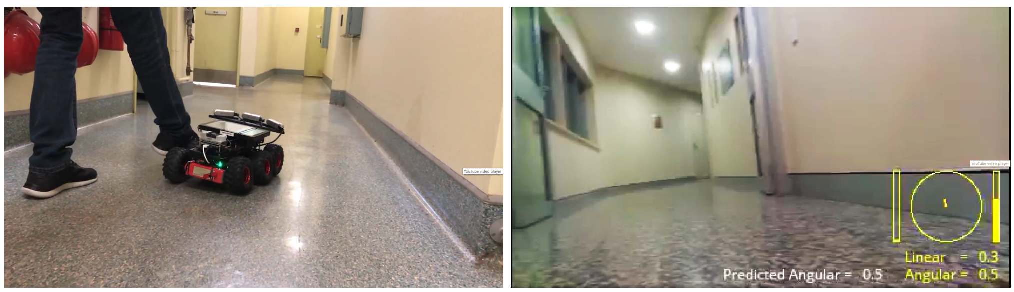

Fig. 2.5. Our indoor Trailnet CNN reacting to external disturbances

THe robot was tested in different hallways, including ones from where we did not collect training data. The robot performed remarkably well in cluttered hallways also, showing its robustness . In addition to that, the response to 90 degree corners was also remarkable, where the robot smoothly turned to follow the hallway.

Fig. 2.6. Our CNN reacting to corners

The results were presented to the fellow scientists in a Robotics Reading Group meeting as a demonstration of the end to end pipeline I designed.

2.3 Building Wallie: A Hardware Platform for Data Collection and Deployment

To collect data, I sketched a small camera rig to hold three Intel Realsense cameras, each facing center, left and right at 30 degree angles from center. Samith Ashan, my coworker helped us by designing it in solidworks. With the help of other CSIRO members, we 3D-printed the camera rig.

Fig. 2.7. Wallie: The Robot

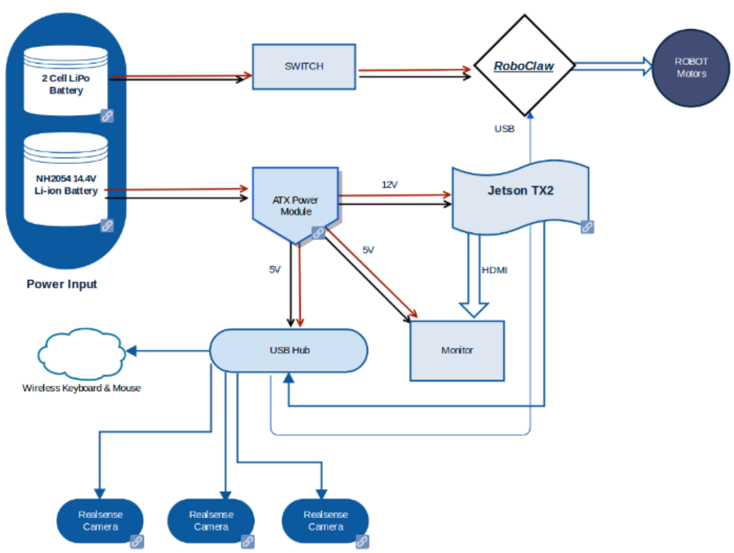

Fig. 2.8. Wallie: Hardware Hierarchy

NVIDIA Jetson TX2

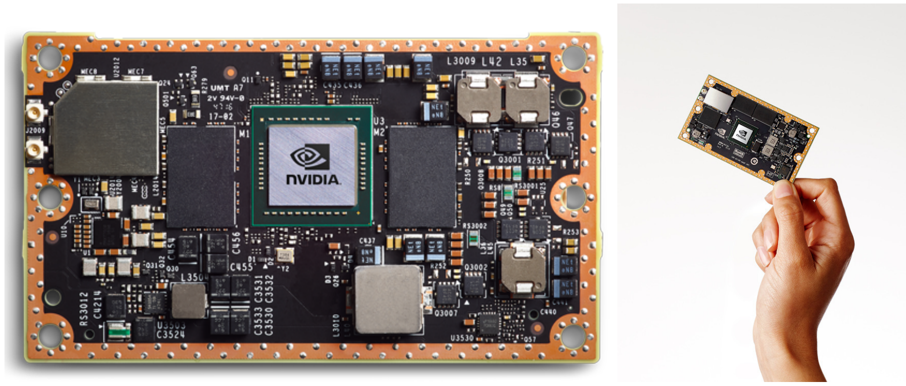

Fig. 2.9. NVIDIA Jetson TX2

Intel Realsense Depth Camera D435

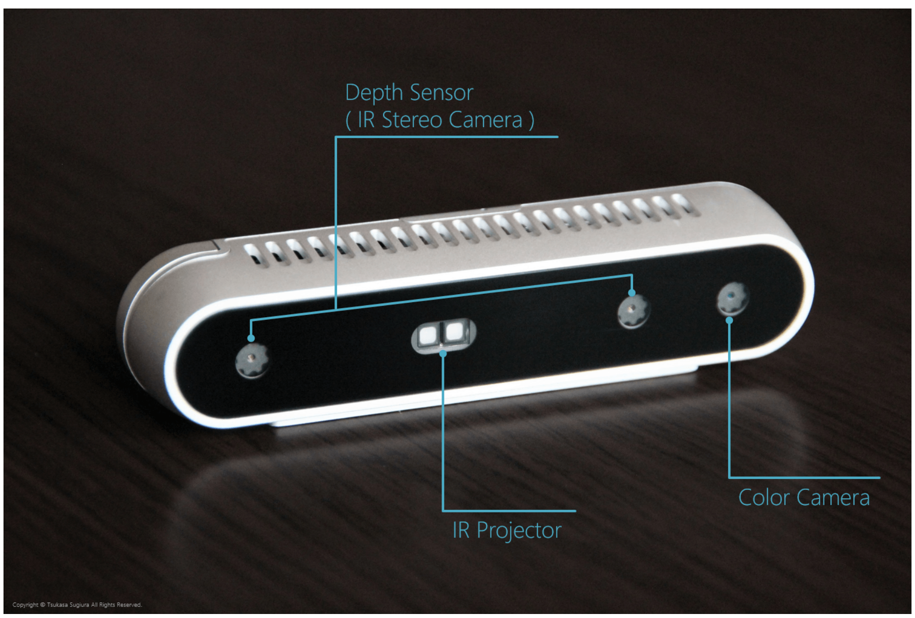

Fig. 2.10. Intel Realsense D435

Roboclaw Motor Controller

Problems Faced and Solutions

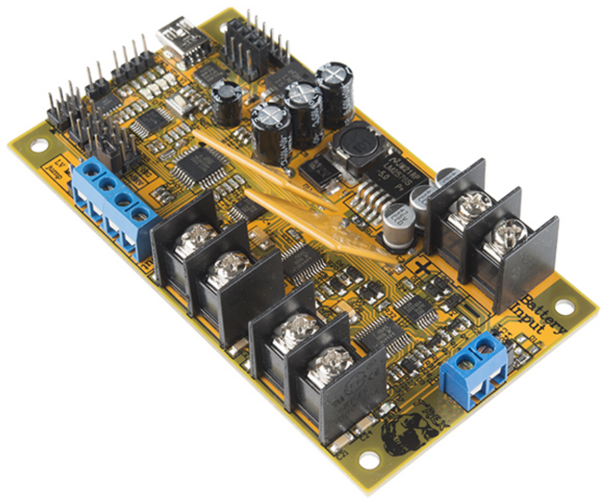

Fig. 2.11. TREX Motor Controller

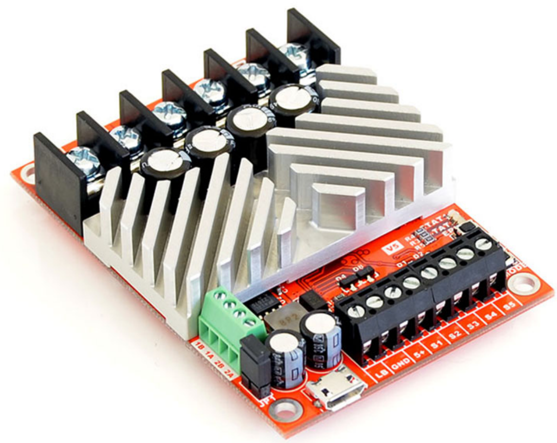

Fig. 2.12. Roboclaw Motor Controller

2.4 Designing and Implementing an Efficient End-toEnd Pipeline for Machine Learning in Robotics

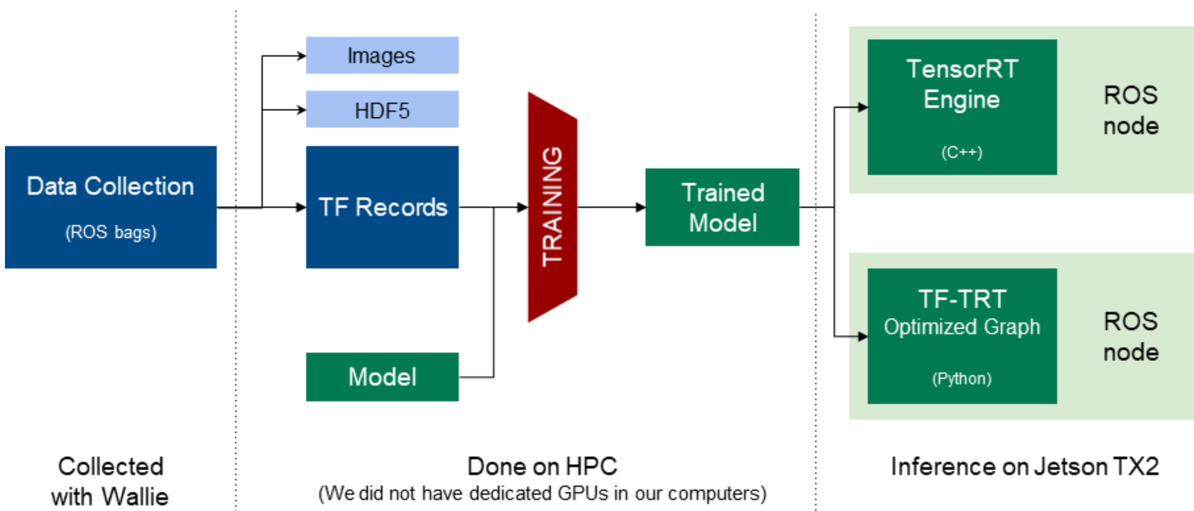

Fig. 2.13. End-to-End Pipeline

2.4.1 Training on Supercomputers

2.4.2 TF Records

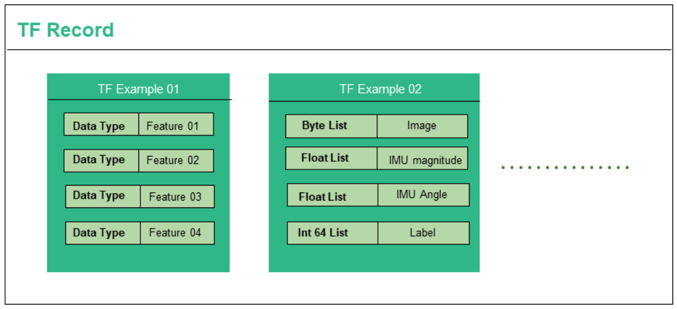

Fig. 2.14. Structure of a TF Record

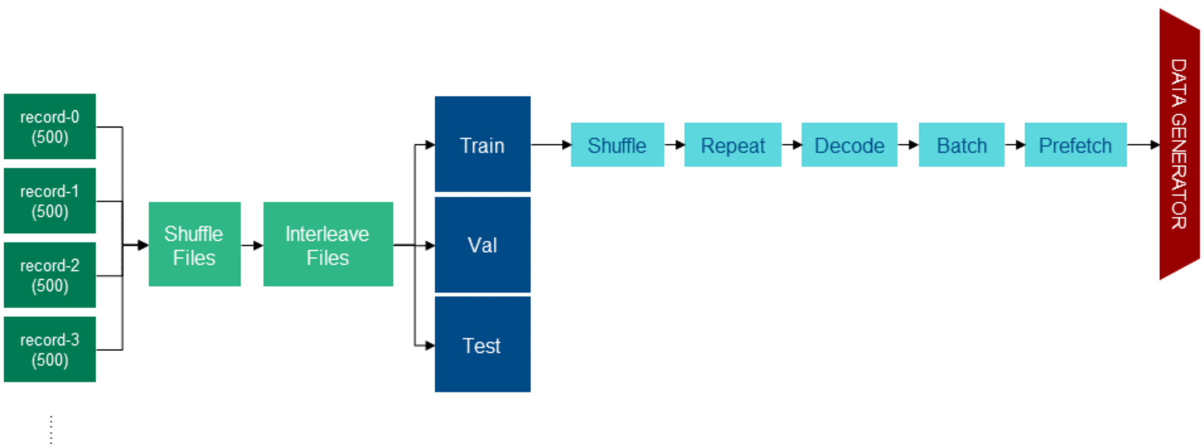

Fig. 2.15. Data Input Pipeline with TFRecords

2.4.3 TensorRT: Deployment on a low power device

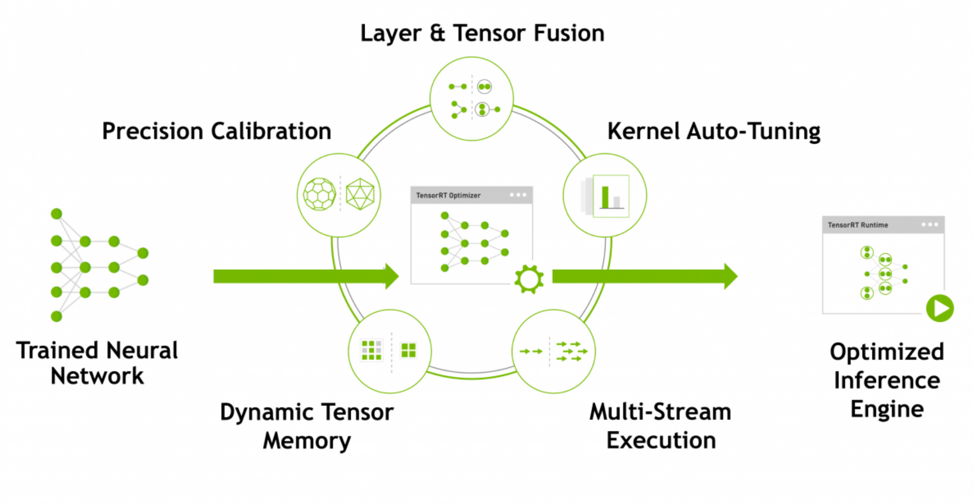

Fig. 2.16. TensorRT in a nutshell

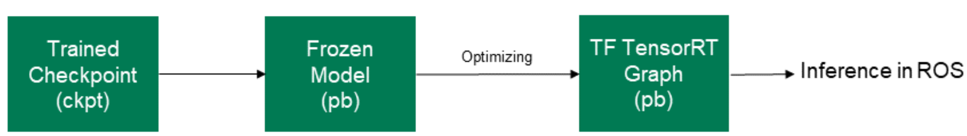

Fig. 2.17. Deployment Pipeline: C++

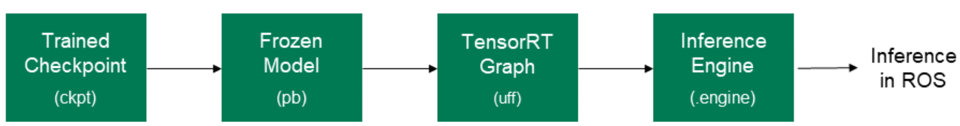

Fig. 2.18. Deployment Pipeline: Python

2.4.4 Problems Faced and Solutions

2.5 Hillnet: An Experimental Attempt at Utilizing ML for Hill Climbing

2.5.1 Preprocessing IMU and Velocity Data

2.5.2 Data Collection

Fig. 2.19. Data collection to train hillnet

2.5.3 Classification Approach

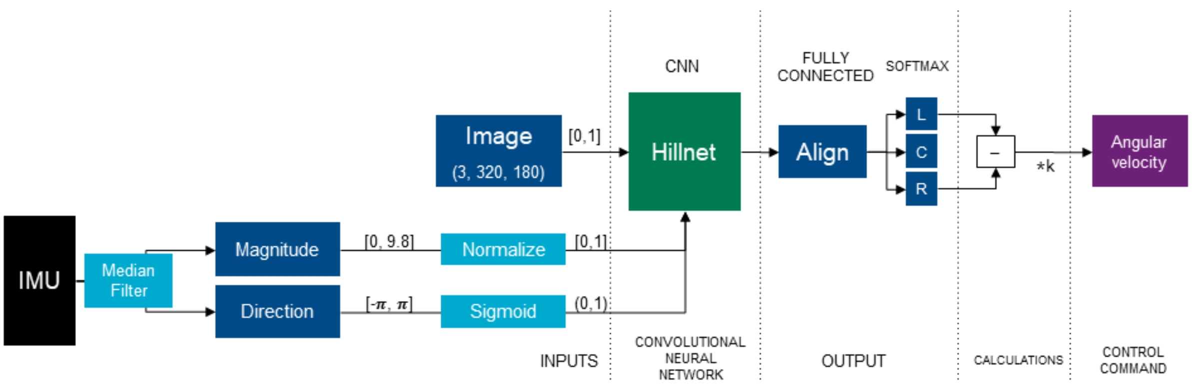

Fig. 2.20. Hillnet Classification Architecture

2.5.4 Regression Approach

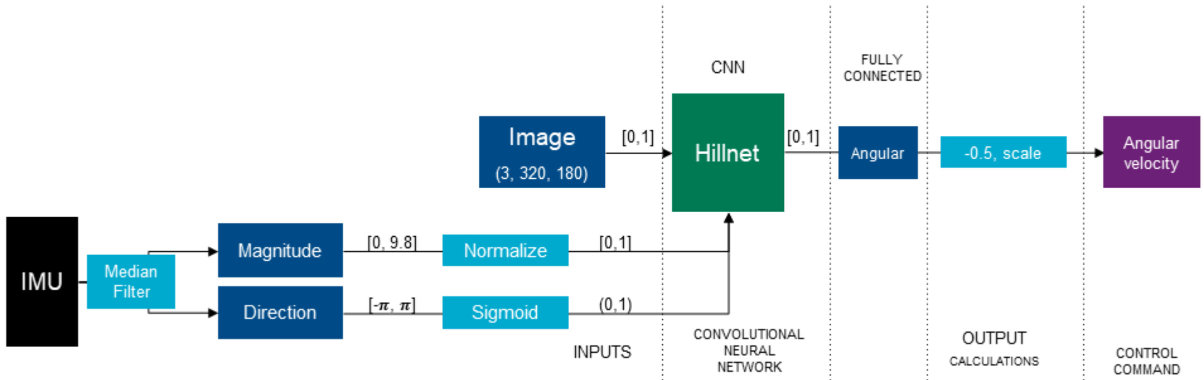

Fig. 2.21. Hillnet Regression Architecture

2.5.5 Merging Scaler and Image Inputs

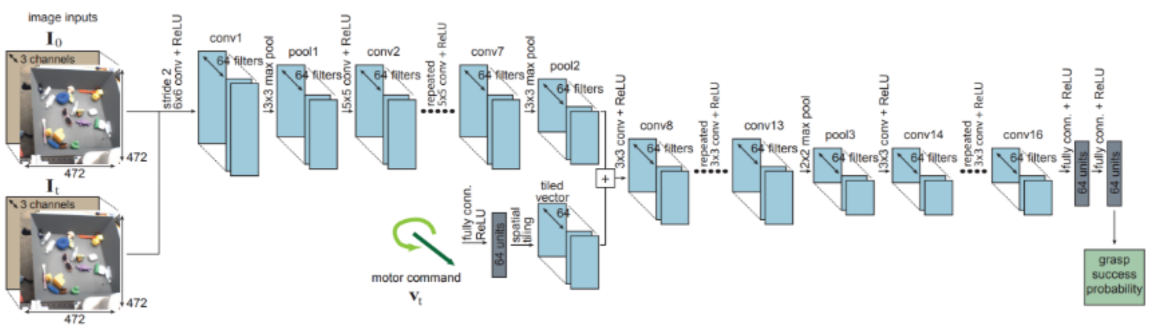

Fig. 2.22. Merging by Broadcast and Add

2.5.6 Problems Faced and Solutions

2.6 Life at CSIRO

2.6.1 Reading Groups and DATA61 Meetings

2.6.2 DATA61 Live Event

Fig. 2.23. DATA61 LIVE Event

2.7 Presenting the Pipeline at Reading Group to the Scientists

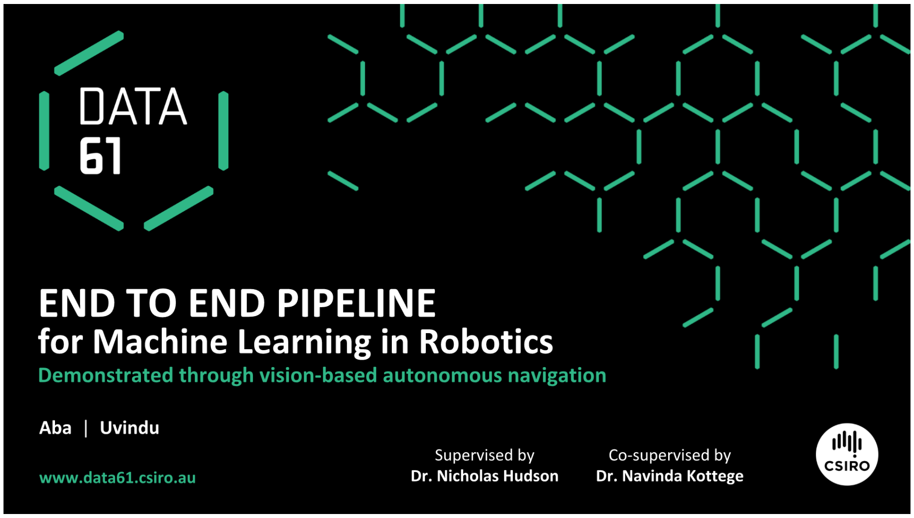

Fig. 2.24. Presenting the Pipeline in Robotics Reading Group

3 Conclusion

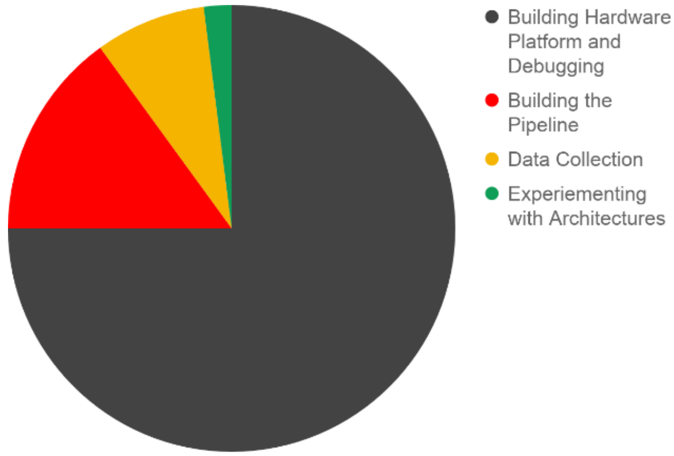

Fig. 3.1. Overview of our time spent

References

[1] Nvidia jetson tx2 module. https://developer.nvidia.com/embedded/buy/jetson-tx2. Accessed: 2018-06-28.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield. Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness. 2017.