Contents

List of Figures

List of Tables

Preface

Described in this report is the 24 week training experience I had in wave computing/Paraqum Technologies dating from 25.06.2018 to 07.12.2018 for the fulfillment of industrial training requirements of my B.Sc. in Electronics and Telecommunication Engineering. The report consists of 3 main sections as follows

Chapter 1: Introduction to Wave computing and Paraqum Technologies

Wave computing is a USA based AI startup and Paraqum Technologies is a Sri Lankan Electronic design startup. Wave is now ranked among the top 25 AI providers of the world [7] and Paraqum Technologies is highly respected in Sri Lanka as a pioneer of the industry and recieves design contracts from all over the world. Our training was done through a design contract offered to Paraqum by Wave computing.

Chapter 2: Training Experience

Even though I was an intern, I was given full exposure to the company workflow. I recieved a full scale project to lead and a support team from all over the world to work with. Everyone in the company were very supportive and friendly. The environment was comfortable to work and the workload was perfectly balanced, demanding yet not overwhelming.

Chapter 3: Conclusion

I can say without a doubt, this is one of the best training experiences one can get. The University of Moratuwa Industrial training division, the Department of Electronic and Telecommunication Engineering and National Apprentice and Industrial Training Authority (NAITA) together did their best to provide us with an exceptional experience that will be very useful for the future of our careers.

Acknowledgment

I take this chance to show my gratitude towards everyone that helped make this training possible and facilitated my time with it. First of all I should thank the industrial training division of University of Moratuwa and the National Apprentice and Industrial Training Authority (NAITA) for allowing us to get an industrial training of this magnitude and quality during the time of our studies and always looking out for the wellbeing of the trainees.

I also would like to thank the administration and staff of Paraqum Technologies, starting with the CEO Dr. Ajith Pasqual, Manager Eng. Hasanka Sandujith, Eng. Kasun Tharaka who coordinated the internships and all other staff members who helped in a great many ways to make the training a staggering success.

Last but definitely not least, The staff of the Wave computing team (later separated as Wave computing Sri Lanka) were always dedicated to make the training worthwhile for us. I sincerely thank the Senior Director of Software Engineering division of wave computing Eng. Henrik Esbenson, who visited and supervised us from time to time, General manager of Wave computing Eng. Nuwan Gajaweera, Technical leaders Eng. Upul Ekanayaka and Eng. Binu Amarathunga, My supervisors Eng. Achintha Ihalage, Eng. Omega Gamage and SDK team lead Eng. Dakila Serasinghe and all other staff members of wave computing for everything they did for us.

Chinthana Wimalasuriya,
Undergraduate,
Department of Electronics and Telecommunication Engineering,
University of Moratuwa.

1 Introduction to the Training Establishment

1.1 Company Overview

Wave Computing is a USA Silicon Valley based company that specializes in dataflow technology [3], an alternative to the tensorflow [2] technology used by tech giants such as Nvidia and Google to accelerate AI and neural net training. Although the company is relatively new to the game, they have had a number of notable achievements, which are explained below in detail. They outsource their design contracts to teams around the globe and one of these contracts was undertaken by Paraqum Technologies. This particular team works with the Software development kit (SDK) and the applications of the Wave dataflow platform.

Fig. 1.1. Wave Computing logo

Paraqum Technologies is one of the very first truly Sri Lankan Electronics industry companies in the country. It undertakes Electronic design contracts from big companies from around the world such as Osprey video and of course, Wave Computing. They also have their own network product line. The company had one of their teams working on a design project of Wave Computing which split up in November 2018 to form the Sri Lankan branch of Wave Computing (pvt) Ltd.

Fig. 1.2. Paraqum Technologies logo

My work was focused on the SDK layer of the Wave dataflow platform. This describes a large toolchain of software utilities that are the primary way that an advanced user would interact with the system. This team is also one of the busiest teams of the whole organization because of the sheer complexity of the SDK. In the context of the Sri Lankan staff, the SDK team is the heart of the whole project.

All details disclosed under this section is intellectual property of Wave Computing. I have included details of the training experience as much as possible while abiding with the requirements of Wave Computing on information disclosure.

1.2 Company History - Wave Computing

Wave computing is primarily a project-turned-to-startup by Dr. Derek Meyer, who is also the current CEO. The idea is to design a processor to process the huge amount of data required for complex operations such as large matrix multiplication applications in parallel. He realized his idea in to a semiconductor manufacturing company, wave semiconductor, which has become a legacy as some urls of the company still reads 'wavesemi'. As the prospect of Artificial intelligence became popular, the company understood that their time and effort is best invested in that field and hence, wave semiconductor reformed into Wave Computing AI.

To meet the demand for Artificial intelligence applications, wave computing has planned a series of devices which can fit the requirement of the users whether it is a large datacenter or a small scale business or mobile/IoT devices.

Fig. 1.3. Wave Datacenter Server Unit

Fig. 1.4. Wave Consumer Unit

MIPS Acquisition

MIPS is one of the old time giants of the silicon design game. To facilitate their ideology of integrating dataflow technology to small devices, Wave Computing recently acquired MIPS [5] for their silicon design expertise. With this step, Wave computing hopes to realize their idea of "Bringing datacenter to the edge" by installing datacenter level high performance hardware in small(edge) devices.

1.3 Company History - Paraqum Technologies

Started in 2013 as a continued final year project of four students from the 2008 batch of University of Moratuwa Department of Electronic and Telecommunication Engineering, Paraqum Technologies was first located inside the university itself. The CEO also being a lecturer in the University, this was the result of the attempt to promote technical entrepreneurs to rise up from the university level.

Paraqum Technologies quickly developed into a full sized technology startup and by 2018, the staff had grown to around 40 and every year they took around 10-12 interns under their wings to properly expose them to the electronic industry. Now they have design contracts from renowned industrial leaders and their own networking product line which is also very famous for their unique capabilities.

Fig. 1.5. Paraqum Technologies Staff (including Wave team members) [1]

1.4 Wave-Paraqum partnership, separation and its effect on interns

As described above, Paraqum Technologies had a design contract from wave computing for their toolchain and application needs. This contract had been in place for almost two years and has proved to be one of the most productive teams wave computing has ever employed. They have helped in designing the hardware part of the dataflow processing chips and by the time we started our internships, they were working on RTL level software projects.

However, Wave computing had decided it would be better to acquire the Sri Lankan team for themselves. So, in november 2018, the Wave Computing team was separated from Paraqum Technologies and established as the Sri Lankan branch of Wave Computing (pvt) ltd. with no connection to Paraqum Technologies whatsoever. Before Separation, Wave computing team worked in the Paraqum Technologies office in Kohuwala but after that they moved to a new office in Bambalapitiya.

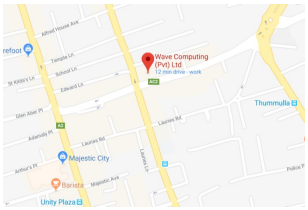

Fig. 1.6. Wave Computing Office location

As the organizations separated, interns of the wave team also moved to work in the new office premises and work was carried out as normal. But the internship contracts were not changed and technically, we were interns of Paraqum Technologies for the whole 24 weeks but worked under the supervision of Wave Computing staff. Therefore this report will be more focused on Wave Computing.

1.5 Organization Structure and Hierarchy

Wave computing team was another division in Paraqum Technologies until the separation but after that they became a fully independent entity. The dotted line in the following diagram represents the administration link that existed before the separation. Now Wave computing Sri Lanka

operates directly under the administration of their head office in Campbell, California.

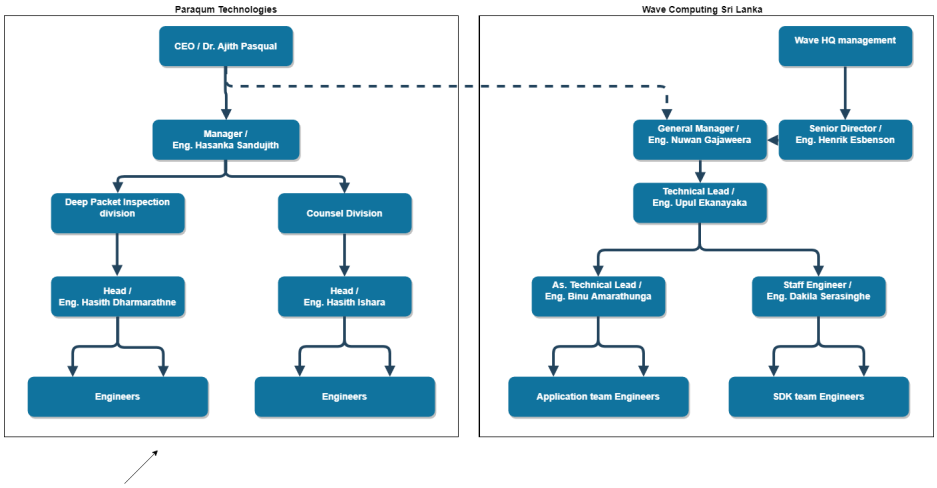

Fig. 1.7. Wave/Paraqum administration structure

1.6 Areas of Interest

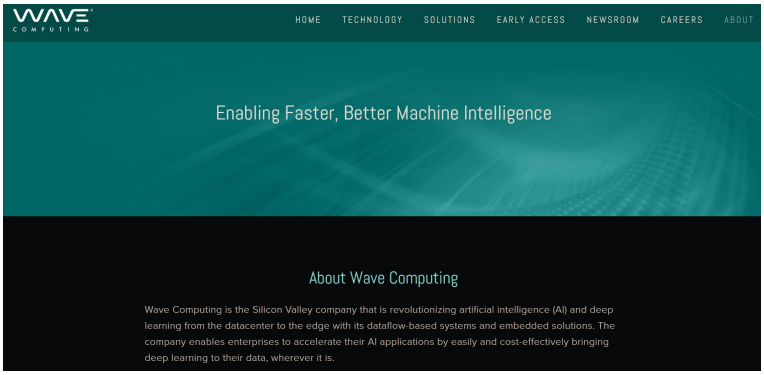

Fig. 1.8. Wave computing Homepage with their target of better AI

The main focus of Wave computing is to Introduce new processing system for heavy computational operations with their proprietary dataflow processing technology. When the system is built and released, it can be used for:

Artificial Intelligence

The main use of the dataflow processing units (DPU) is intended to be Artificial Intelligence systems. Current hardware in our devices are not suited for the massive amount of data that need

to be processed in parallel. The DPU has been optimized for this particular task to perform it at high speed. It is speculated that once wave DPU systems become available in market, it will spark a revolution in the AI industry.

Edge devices

Edge devices are small devices such as mobile phones and IoT sensors. Wave computing already has plans in motion to develop high performance, small size hardware to be installed in these edge devices. This will allow us to have very powerful yet small devices for our day to day usage.

Image processing

Image processing and AI are similar in many ways. The parallel data handling capacity of DPU systems makes it an ideal candidate for image processing applications such as self driving or biometric authentication.

1.7 Current Situation

The initial release of Wave hardware to the general public is scheduled for 2020 and the company is doing well on the roadmap to meet this target. The hardware design is nearly perfect and the software is halfway through. It can be expected that the company will meet the target soon enough and they are expanding rapidly now, with the new office in Sri Lanka hiring more and more people.

1.8 Impacts on Sri Lankan Industry

Having recently completed a very successful series E funding round [8], Wave Computing has a fairly large capital at its disposal and are ready to invest well in the Sri Lanka division. This huge amount of money will directly enter Sri Lankan economy in US dollars, aiding the stabilization of economy. They also offer employment to talented local engineers in bigger and bigger numbers every year. Finally, when the Wave product line enters market and becomes a key element in the AI game, Sri Lankans will have an important role to play in it, carrying us forward in the technical development sector.

1.9 SWOT Analysis

Strengths

- Wave already has MIPS under them, giving them a considerable head start on the silicon game. They got a team of experienced engineers with this acquisition.
- They have teams around the world with different perspectives, allowing creative solutions to problems.
- The company is very well funded, giving them plenty of runway.
- The startup culture is well maintained in the workplaces, making it appealing to employees.

Weaknesses

- Without any kind of product on the market yet, the company solely runs on VC funding
- Global teams sometimes cause delays in communications.

Opportunities

- AI field is expanding day by day, meaning the potential market for the company widens continuously.
- Another funding round can result in more and more funds due to the promising nature of the project.
- Reputation of the company is luring in more talent as it expands.
- Since the hardware is not restricted only to AI applications, there can be more potential uses to the devices.

Threats

- There may be rival companies with better solutions to the AI processing problem than the DPU technology.

- Some of the global teams are from politically and economically unstable countries and they might be lost to national crisis.
- DPU is not fully built yet. It may not give the expected outcomes and performance in real world.

1.10 Suggestions to Improve the Company

- Some sort of outbound activities can be organized to allow the team to bond even better.
- Interns could be provided with some more training sessions.

2 Training Experience

2.1 How I got the Opportunity

A long time before the industrial training selection, I had heard Paraqum Technologies was one of the best places to learn about the electronics industry in Sri Lanka and Its CEO, Dr. Ajith Pasqual had taught a module in the university that sparked an interest in me about the subject of silicon design. Therefore, when Paraqum Technologies was listed as an open CV company, I did not hesitate to submit a CV, which got selected by the company staff who then interviewed me thoroughly in their office and sometime later, informed me that I had been selected to the Wave Computing Division.

At the start of the Internship, I was placed under the supervision of Eng. Achintha Ihalage, an application engineer whose original task was to handle the timing and constraints of the DPU chip. He walked me through the basics of setting up the Wave Computing workspace, company work ethics and other needed technical skills. He also introduced us to the DPU hardware and other proprietary technologies by Wave.

During the second week, Eng. Henrik Esbenson, who is in charge of the Sri Lankan team at Wave HQ visited the office and demonstrated his ideas for projects. One of these was the Python - Wave Flow Graph translator, also known as Py2WFG. I volunteered to take that project and Eng. Henrik allocated me the necessary resources of the company including support teams and software tools to carry on the project.

This project soon became popular among the crowd of wave computing and I developed it according to the requirements and feedback. The amount of work was rather large but I managed to complete the project and hand it over by the time Internship ended. This project will most likely be adopted by full time developers and expanded to probably replace the existing design flow.

2.2 The Teal Architecture and Wave Flow Graph(WFG)

Teal is essentially the best of both programmable logic platforms and Application Specific Integrated Circuits(ASIC) combines together to give out the best performance and power efficiencies. In fact both hardware and software designs can be transferred onto the Teal fabric. Basically, Teal can be broken down into smallest unit of a Processing Element(PE). This is a simple processing unit which can carry out 8 bit operations adhering to Reduced Instruction SET (RISC) architecture.

2.2.1 DPU Architecture

A Processing Element will inherit an 8 bit accumulator who in return is connected to its neighboring Processing Elements. It is evident that a combination of a large number of such Processing Elements achieve as much parallelism as witnessed in the computational design history of the world. This revolutionary design with extensively parallel computational ability is a completely ground-breaking approach to the existing technologies used in chips for parallelism. Incidentally, a combination of 16 Processing Elements make up for a Cluster and a combination of 64 Clusters form a Super-Cluster and 16 such Super-Clusters form a single Wave Dataflow Processing Unit. The projected design for this massively complex structure has now reached the taping out process an it will only be a matter of time before the first chip gets manufactured physically. Such a developed Processing Element is projected to have a speed of close to 10GHz.

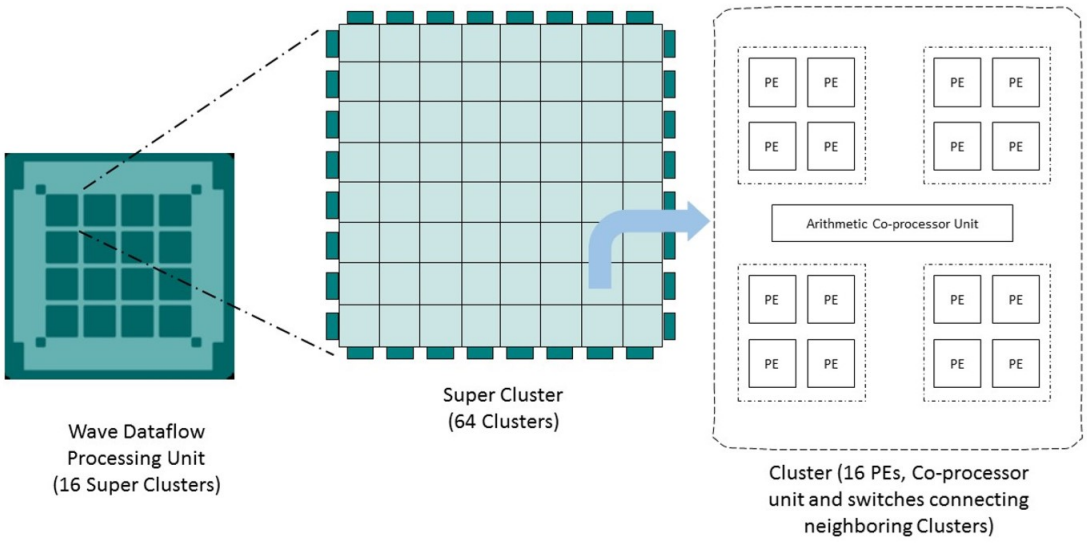

Fig. 2.1. Wave DPU with Processing Elements(PEs)

The Figure 2.1 is a comprehensive illustration of the structure of Teal design. It is noteworthy to realize that each Processing Element acts much similar to a single processing unit and therefore, each Super Cluster can be viewed in the perspective of thousands of processing units ready to function simultaneously. Nevertheless, the true power of Wave Dataflow Processing Unit design is shown in Figure 2.2 where the extent of operation of the final design is depicted. The final design is projected to have a whopping 2 Peta Operations per second amount of power which is ideally suited for the needs of computational power of the future. It is safe to say that Wave Dataflow Processing Unit will become by far the fastest ever processing unit of the world.

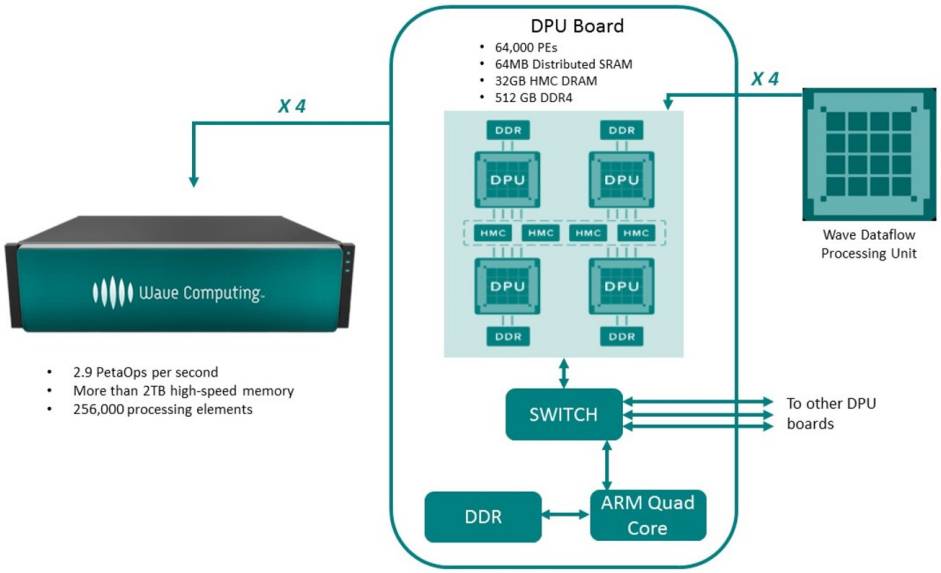

Fig. 2.2. Wave Deep Learning Computer

2.2.2 SoC Perspective

The Teal processing accelerator is essentially integrated into a System-on-a-Chip(SoC). As shown in the Figure 2.3 it is connected to processor SoC using streaming AXI interfaces. It has a direct connection toward high speed IO streams. The DMA controller (refer Figure 2.3) is the key connector between external DRAM internal Block RAMs of Teal. The accelerator itself is demand driven, meaning it will stay at sleep state unless instructed otherwise. The importance of this viewpoint was important to me as some of the design work carried out were finding a way to simulate DMA inputs and outputs. Since accessing data from external RAMs is through DMA 'Channels', some of the work related to improvements in the DMA engine were carried out during

my project executions.

The next interesting design feature of Wave Data-flow processing model are the Wave Flow Graphs. This is a representation scheme for which operations are represented as nodes and values as edges. Nodes and edges connect together in a network of operations to perform a computational task. The idea behind the clock-less operation of the tasks come through the ability for nodes to perform the intended operation whenever valid inputs are provided. A dedicated clock is not needed and thus presence of valid data at the inputs automatically will define trusted operation.

However, a sense of timing is available in the form of the concepts of tics and sub-tics. A tic is equivalent to a circular round of operations executed by a Processing Element buffer which is currently defined as 256 instructions. In fact, it is 256 times the instruction cycle time. Due to the fact that instruction cycle time is not specifically designated but rather dependent on the execution of each operations' processing speed, it is not fair by the Wave DPU design to have a specific tic-rate. Incidentally, there is a typical rate of 10 GHz for a sub-tic cycle. a sub-tic cycle is essentially the rate at which an instruction from a circular buffer(IRAM) is fetched, processed and switched.

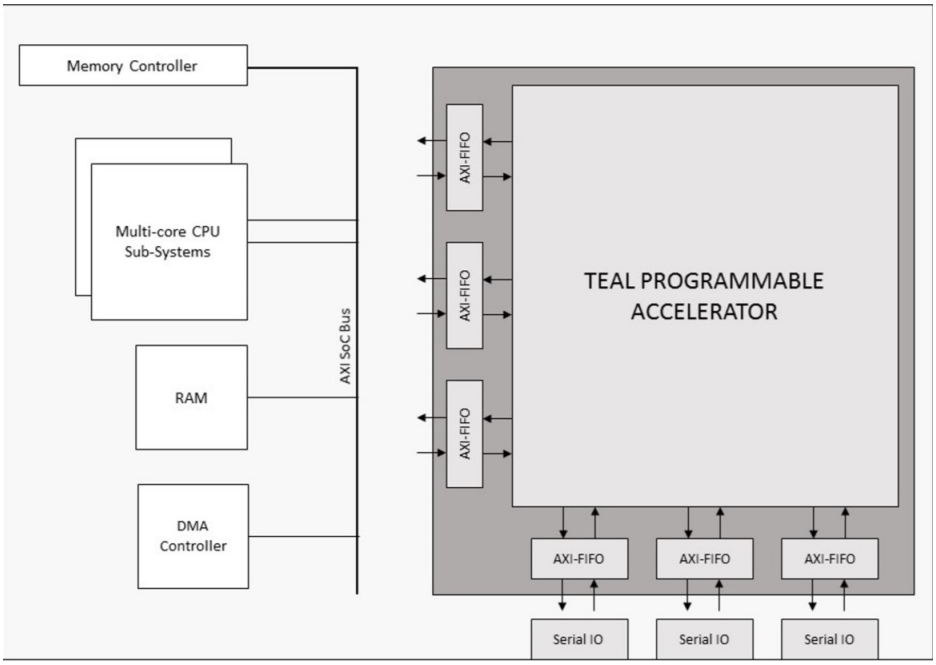

Fig. 2.3. SoC Perspective of Teal Programmable Accelerator

Apart from the contrast between the parallelized implementation of Wave Flow Graph designs and the continuous flow of operations within PEs, Wave Flow Graph is a defined rule to represent any program within the Teal fabric. In XXXXXXX section 1.3.3 (Unit Testing) XXXXXX a direct implementation of testing the Wave Flow Graph operations carried out by me, is explained in detail.

2.2.3 Wave design flow

Owing to the complexity of parallel PE operation it is evidently clear that programming instructions into the Teal Architecture is not achievable manually. Therefore, an EDA flow is prevalent so that any hardware design can be brought down through compilation, scheduling, mapping and routing to the Teal fabric level. This flow of compilation from hardware design to Byte-fabric level is done through the Wave introduced design flow. The Figure 2.4 is a representation of a very simple Wave Flow Graph design where you can see the nature of operation of the language. It basically consists of bit wise operations between inputs and outputs. Most operations are related to bit level manipulation of data and it is note worthy to realize that these operations are carried out for 8 bit data chunks. The combination of several bytes of data can be processed with special modules of Wave Flow Graph code where the data is subdivided and processed accordingly. The node edge representation of the design will give you enough ides as to the complexity that can arise with the introduction of several operations and variables into one design. The final structure will be a collection of various nets combined together from input end to output end.

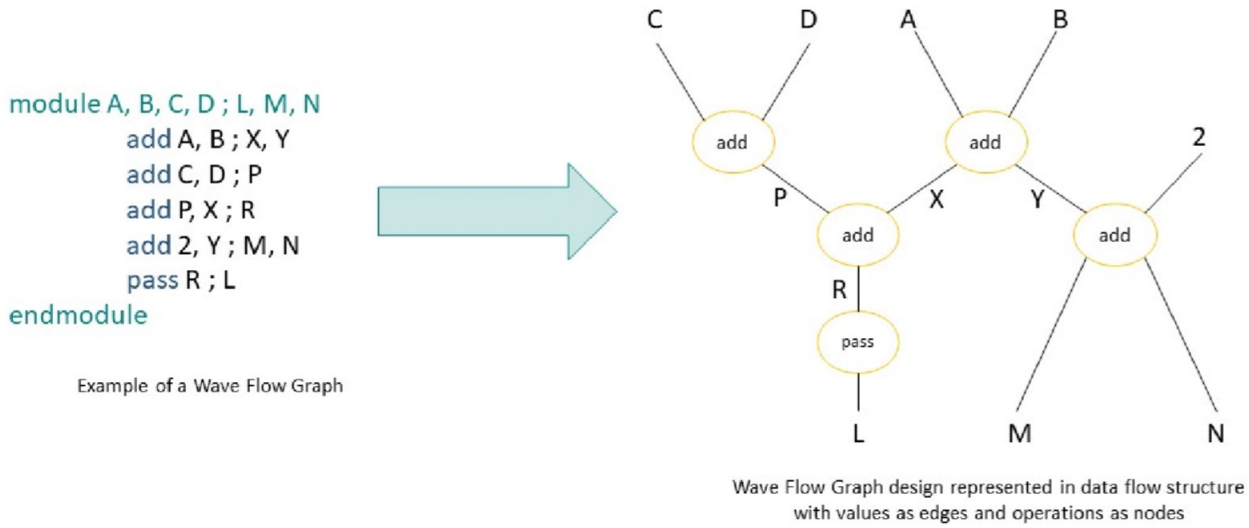

Fig. 2.4. Data flow structure of Wave Flow Graph designs

2.3 Dive Framework and Wave toolchain

2.3.1 Framework Overview

When writing a new design for the DPU, the engineers use an in-house written API called the dive framework for testing and Verification. Due to the sheer complexity of the programs that need to be adapted to run in the DPU, the code is brought down to lower semantic levels step by step using the toolchain which consists of 5 main tools.

- Wave C Compiler (WCC) - Compiles the C++ code to a WFG code
- Wave Flow Graph Simulator (WFGsim) - Simulates the flow graph and compares it with the I/O values of the C++ code
- Wave Flow Graph Compiler (WFGC) - Compiles the WFG code to a Wave Assembly code
- Wave Machine Simulator (WMsim) - Simulates the Wave assembly code in a virtual machine environment
- Wave Assembly Compiler (WAsm) - Compiles the assembly code to an encrypted binary file format called Lantana

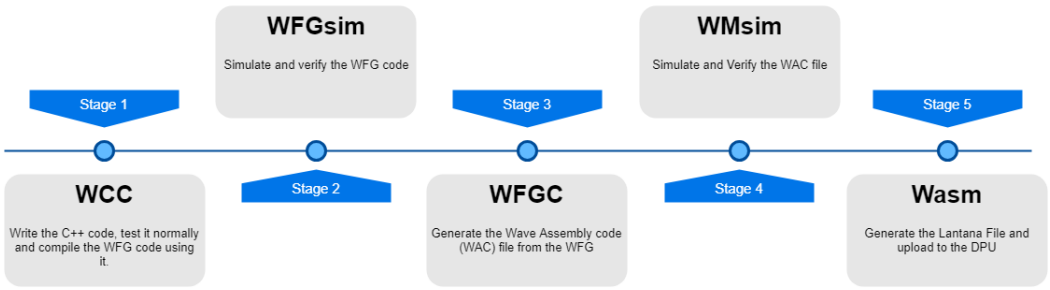

Fig. 2.5. Wave Toolchain and design flow

2.3.2 Repository Structure

Different cloud folders that contains software code are known as repositories. Wave source codes are stored in five main repositories hosted in a local git server. For better identification, the tools are again divided in to two classes, Wave Front End (WFE) and Wave Back End (WBE). WCC, WFGsim and WFGC belong to the WFE while WMsim and WAsm are contained in the WBE. In the Wave source code, the code for WFE and WBE are stored in separate repositories while the code that is shared by these are stored in a third repository named wcore. Test codes are stored in the wtest repository while experimental tools are stored in the final repository, tools.

2.3.3 WCC

Wave computing has their own version of C++ dubbed WaveC, which is C++ with added support for data channels and some more things. This language receives constant updates from a team inside wave and thus, developers must build the C language from its source code frequently. This is also true for other tools.

WCC uses a technology called LLVM [4] to convert one human written code in to a different human readable code. It works by analyzing the code and simplifying it to a desired level and assembling the result back together on a predefined syntax. After simplifying the code again and again, the desired output can be obtained, which in this case is the WFG code file.

2.3.4 WFGsim

This was the most important tool for me since my project could eventually evolve to fully replace this tool. I had to analyze this tool very thoroughly to extract functionality for the Py2WFG

simulator (section 2.4.3).

WFGsim works by multi-thread programming on python. It initializes separate threads for DPU sections and host processors. Inside each python thread, a library called boost is used to invoke a linear C++ script. These scripts 'talk' to each other via the python interface and python interface also gives the DPU sections a sense of time by governing the order which operations are executed.

2.4 Py2WFG: A better way to write Wave Flow Graph

2.4.1 Python Vs. WFG

```python
class job():
    def __init__(self, payment, start, end):
        self.payment_ = int(payment)
        self.start_ = int(start)
        self.end_ = int(end)

def get_end_day(jobs):
    res = 1

    for j in jobs:
        if j.end_ > res:
            res = j.end_

    return res
```

Fig. 2.6. Typical Python code snippet

The Figure 2.6 shows a typical python code example and Figure 2.4 shows a WFG snippet. These two languages were built for two entirely different purposes but the highly adaptable nature of python makes a valid point whether if it can replace the functionality of WFG. But both languages have their pros and cons.

Python

Python is a general purpose scripting language which focuses on user friendliness. It supports object oriented programming and is also backed up by a huge number of highly optimized libraries for various purposes. But when compared with languages such as C++ and Java, Python is heavy on the memory and slow for large volume computing. It is also not optimized for the particular purpose of programming the Wave DPU.

WFG

WFG was created with one purpose and one purpose only in mind, Programming the Wave DPU. Thus it has primary operators that can precisely match the deep capabilities of the DPU hardware. It can be very efficient too when properly programmed. The downside to this language is that it is very strongly typed and is a user friendliness nightmare. Repetitive operators need to be manually entered by the user and the language has no capability of any kind of looping of the language itself.

2.4.2 Python to WFG translator(Py2WFG)

A solution was created by putting the best of both worlds together. The idea is to create a python 'sleeve' to cover up the WFG interface and present the user with a way to interact with python and get WFG level results. The user will now write up the script and when he runs it, it will output a fully optimized WFG script.

Fig. 2.7. Part of the Py2WFG source code

Code Translation

The basic idea for this library is derived from Tensorflow [2], which is a python library that allows easy neural net design using a python library. The user will still need a sufficient knowledge on WFG syntax to use the library. But it will eliminate the hassles of writing a slightly-above-assembly language scripts by hand. The script that user needs to write is similar to the one showed in Figure 2.8. The library needs to be imported in to the script, Then Modules can be created from the wMod class in the library. These module 'husks' are then filled with dataflow operations as needed. When complete, issuing the makeWFG command will write a full WFG script that can be later integrated into the existing wave design flow.

```python
import wave.py2wfg as p

dt_16_k = p.wMod('dt_16_k')

din = dt_16_k.AddBus('din', 16)
dout = dt_16_k.AddBus('dout', 16)

inputs = dt_16_k.Input(['full', 'empty'] + din)
outputs = dt_16_k.Output(['get', 'put'] + dout)

xor1 = dt_16_k.wxor('full', '0x01', 'not_full').name('checkfull')
xor2 = dt_16_k.wxor('empty', '0x01', 'not_empty').name('checkempty')

and1 = dt_16_k.wand('not_empty', 'not_full', 'put').name('checkget')

or1 = dt_16_k.wor('0x00', '0x01', 'get')
or2 = dt_16_k.opBank('or', 16, [din, '0x00'], [dout])

for i in range(16):
    or2[i].name('dtran'+str(i))

dt_16_k.makeWFG('dt_16_k')
```

Fig. 2.8. Example Py2WFG script

Running the example script shown in Figure 2.8 will output a WFG file that contains the operations represented in the script, which is shown in Figure 2.9

```
'module dt_16_k full, empty, din_0, din_1, din_2, din_3, din_4, din_5, din_6,
din_7, din_8, din_9, din_10, din_11, din_12, din_13, din_14, din_15; get, put,
dout_0, dout_1, dout_2, dout_3, dout_4, dout_5, dout_6, dout_7, dout_8, dout_9,
dout_10, dout_11, dout_12, dout_13, dout_14, dout_15

      'xor full, 0x01;  not_full : name=checkfull
      'xor empty, 0x01;  not_empty : name=checkempty
      'and not_empty, not_full;  put : name=checkget
      'or 0x00, 0x01;  get
      'or din_0, 0x00; dout_0: name=dtran0
      'or din_1, 0x00; dout_1: name=dtran1
      'or din_2, 0x00; dout_2: name=dtran2
      'or din_3, 0x00; dout_3: name=dtran3
      'or din_4, 0x00; dout_4: name=dtran4
      'or din_5, 0x00; dout_5: name=dtran5
      'or din_6, 0x00; dout_6: name=dtran6
      'or din_7, 0x00; dout_7: name=dtran7
      'or din_8, 0x00; dout_8: name=dtran8
      'or din_9, 0x00; dout_9: name=dtran9
      'or din_10, 0x00; dout_10: name=dtran10
      'or din_11, 0x00; dout_11: name=dtran11
      'or din_12, 0x00; dout_12: name=dtran12
      'or din_13, 0x00; dout_13: name=dtran13
      'or din_14, 0x00; dout_14: name=dtran14
      'or din_15, 0x00; dout_15: name=dtran15

'endmodule
```

Fig. 2.9. Output of the Py2WFG code from Figure 2.8

This output WFG script unlike the handwritten scripts, is properly formatted and can be reconfigured easily through the python script again. This ease of use further improved with the introduction of the Simulator.

2.4.3 Python to WFG Simulator

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

After successfully demonstrating indoor-Trailnet built from scratch, trained and deployed with my pipeline, our supervisor Nick described about his idea of experimenting with a algorithm to climb hills while avoiding obstacles using computer vision. We were asked to come up with a system where two different types of inputs are merged: scaler inputs representing the direction of the slope and the RGB image input from a camera to output a velocity command to control the motor controller. Trailnet, which itself was based on resnet-18 was chosen to be modified to build such a neural network.

2.5.1 Preprocessing IMU and Velocity Data

The LORD Microstrain IMU sends its data through serial to its ROS package, which publishes the IMU readings in two types: as a quaternion and set of cartesian x,y,z vector components of perceived acceleration. I decided to use the vector components to calculate the magnitude and direction of the gravity vector projected on the horizontal plane of the robot. Direction $\theta \in [-\pi, \pi]$ was measured with respect to the heading direction and then converted to $[0,1]$ range using the sigmoid function to match the range of the other normalized inputs. I chose to output the angular velocity as a float $\in [0,1]$ using a sigmoid output node (in classification approach) and scale it to the necessary angular Velocity.

$x, y, z =$ Magnitude of the cartesian components of perceived acceleration. x: forward, z: vertical

$r, \theta =$ polar components of the acceleration vector projected on the horizontal plane of robot

$r_{in}, theta_{in} =$ processed values given as input to the network

$$r = \sqrt{x^2 + y^2}$$

$$r_{in} = \frac{r}{g} \qquad \text{normalized by the maximum: g}$$

$$\theta = \arctan(\tfrac{y}{x}) \qquad \theta \in [-\pi, \pi]$$

$$\theta_{in} = \text{sigmoid}(\theta) \qquad \theta_{in} \in [0,1]$$

2.5.2 Data Collection

As per the instructions of our supervisor, Uvindu collected data in the slopes around CSIRO campus during the last few weeks of the internship. He placed the robot on the bottom of the hill and drove the robot following the steepest ascent. When he faced an obstacle, he went around the obstacle and continued to follow the steepest ascent. Image streams from the camera, IMU data and the odometry reading from the motor encoders were recorded in ROSbags.

Fig. 2.10. Data Collection to Train Hillnet

2.5.3 Merging Scaler and Image Inputs

For this task, it was necessary to decide how the two types of inputs: scaler and image are combined in the neural network. Our supervisor proposed a method, where the IMU values are concatenated to the flattened output of the convolution layers, followed by few fully connected layers. The inspiration for this idea comes from the insight that deeper layers of a deep neural network identify higher level features and therefore the flattened average pooling output of the CNN should be containing information about by how much should the robot turn to avoid the obstacle. Therefore, concatenating the IMU inputs, which also tell by how much the robot should turn to follow the hill and following it with few fully connected layers might result in the network learning an OR operation between two inputs.

However, a research paper on merging these kind of inputs [9] proposed an alternative method, where the scaler inputs are broadcasted into a matrix with the right size, processed by few fully connected layers and added elementwise to the output of an intermediate layer in the CNN. The insight behind this is the fact that the output of the fully connected layers might act like a mask

on the image, effectively clouding and directing the decision process of the CNN. After some consideration and experimentation, we decided to follow our supervisor's method since it was more suitable for our task.

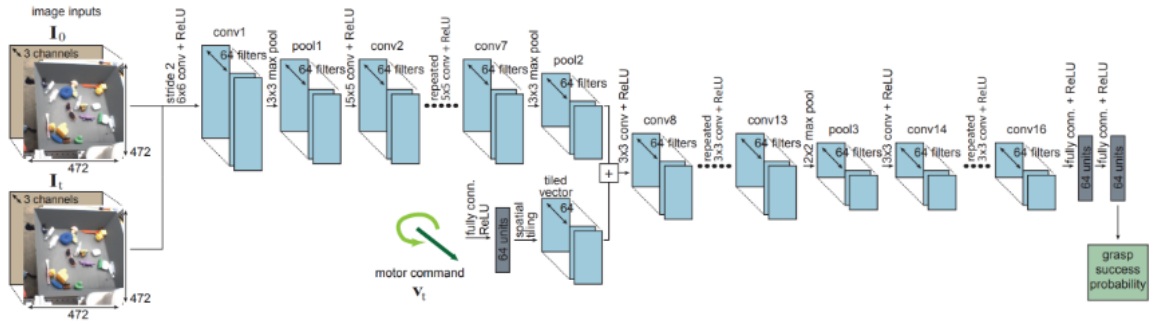

Fig. 2.11. Merging by Broadcast and Add Elementwise as a Mask

2.5.4 Regression Approach

Next we brainstormed on how to post-process the output of the network to steer the robot. During the discussion, our supervisor first suggested to use the regression approach. That is, to build a network with a single output node that has no activation function, so the output value can be directly used as the angular velocity command to steer the robot. This is quite straightforward to build, train and test. The neural network can be trained with IMU and Image data and output as the angular velocity from odometry reading when driven by remote control during data collection.

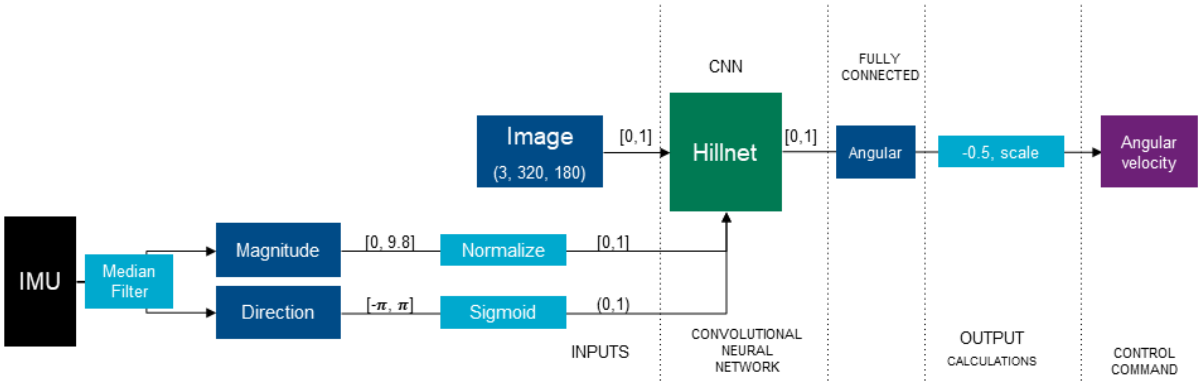

Fig. 2.12. Hillnet Regression Architecture

2.5.5 Classification Approach

During that discussion, I suggested trying a classification approach similar to that of trailnet. First advantage was the ability to fine tune the output by adjusting the constants. With regression, either it works or not. But with classification, we could fine tune it to work as we wish. Then, the effect of human error introduced in collecting data can be made insignificant by quantizing the angular velocity into classes: left, center and right.

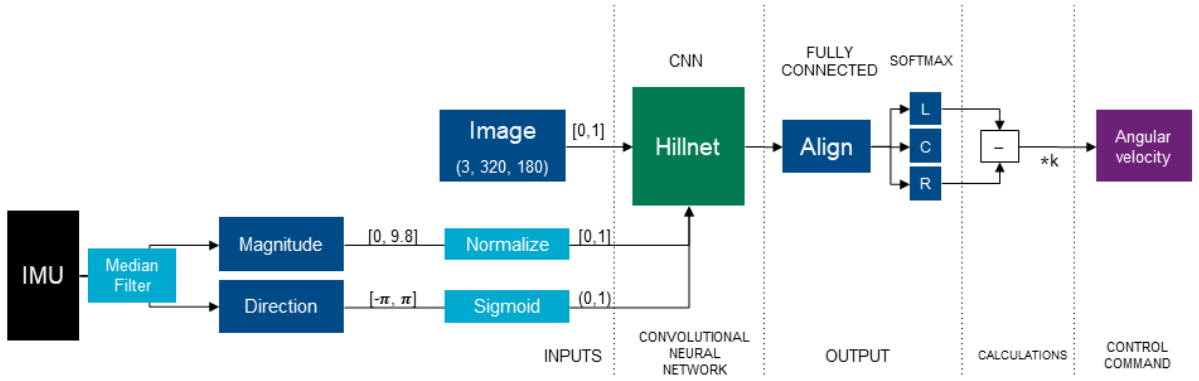

Fig. 2.13. Hillnet Classification Architecture

Next, I proposed a scheme of augmenting data to effectively triple the amount of collected training data. That is, first we record image streams from all three cameras. Then, in the input pipeline, I made changes to add +30°and -30°to the IMU angle and use it as datapoints along with the image streams from left and right cameras respectively. This would train the network to NOT turn away from the steepest slope, unless an obstacle is encountered, I argued. Our supervisor accepted the idea and asked me to work on both classification and regression in parallel to compare their merits.

2.5.6 Problems Faced and Solutions

First problem we faced was the human errors in data collection. Since the hills had a small slope, Uvindu had difficulty in visually identifying the highest-slope direction and steering the robot towards it. When we analysed the collected data using the visualization techniques, we found there was a steady state error by up to 5°-10°. We tried collecting more data with a couple of days remaining to end the internship.

Another anomaly I observed from the visualization was the fact the IMU input had a high variance (noise). This was due to facts the slope was gentle (high percentage error) and the spring loaded suspension system of Wallie was causing it to wobble, introducing low frequency vibrations. This was critical, since our neural network does not remember nor correlate with the past inputs, but considers only the current inputs to give instantaneous outputs. Such a noisy input will prevent the training process from converging to a global maxima. Hence I suggested mounting the IMU at an angle to reduce the percentage error. After experimenting with mean and median filters of various lengths, I implemented a median filter of length 50 to smoothen the noise.

On the last few days of the internship, while debugging the network, I accidentally noticed that our collected dataset had an unhealthy disparity. That is, only 0.15% data accounted for avoiding obstacles, while the rest 99.85% accounted for climbing following the steepest hill. This is a classic problem in data science where the neural network simply learns to suggest "go forward" and be right 99.85% of the time! I had discussed a similar potential problem with the supervisor at the beginning of the project, where I raised concerns that "we are not showing the robot which input-output combinations are wrong. we are showing only what is right". Our supervisor assured that "the robot will learn what is wrong, when you turn the robot to face the hill after avoiding an obstacle". However, the percentage of that kind of data was dwarfed by the "straight climbing" data in the dataset. I discussed this with Micheal and our supervisor and started implementing my idea of artificially boosting the frequency of "avoiding obstacle" data in the input pipeline.

However, we were running out of time by the end of the internship to try all possible ideas and experiment with all the possibilities. I stayed for multiple days overnight at office to try and finish as much as possible, but we couldn't try everything within that short time. Spending most of our time on building and debugging the robot platform could be one of the reasons for our time being limited for exploring new ideas with Hillnet. However, we realized that this is quite common in experimental field robotics, where scientists get to spend most of the time struggling with the hardware issues.

2.6 Life at CSIRO

CSIRO, being a world class research institute, thrives to create a stress-free work environment that encourages people to socialize and to boost creativity. There is a workplace culture in DATA61 to bring cakes (or any equivalent sweets) for all the coworkers if one gets married, arrive at CSIRO, leave CSIRO, has a birthday and so on. I shared Sri Lankan sweets (sent by my mother in mail) for my birthday and cakes with everyone for arriving at and leaving CSIRO.

2.6.1 Reading Groups and DATA61 Meetings

Every friday, a small meeting called "Robotics Reading Group" is hosted, where one scientist explains his current project to everyone who attends the meeting. This way, we get to know the latest technology that is being developed in different parts of DATA61 and new projects that are being started. This meeting also allows the scientist to be questioned, so that he can derive insights from the audience and refine his procedures in the future. Also, once in a fortnight our supervisor Nick holds a "ML Robotics" meeting for all engineers working on machine learning. There we discuss our current ideas and issues to help each other. In addition to these, there are monthly meetings with the entire CSIRO (branches from all over Australia join via video conferencing), where new developments are discussed. One such meeting had the lead scientist from NASA's Insight Mars Lander mission as the chief guest explaining the challenges faced in their mission and taking our questions on the matter.

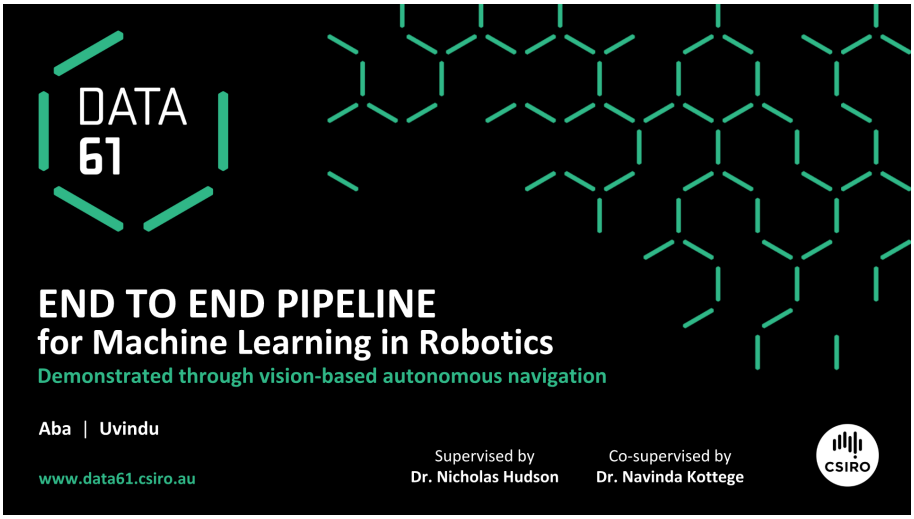

Fig. 2.14. Presenting the Pipeline in Robotics Reading Group

2.6.2 Presenting the Pipeline at Reading Group to the Scientists

Uvindu and I presented [6] the pipeline to other scientists during one of the last Reading Group meetings of the year. It was well received, thanks to the support of our supervisor who encouraged others to use our pipeline in their workflow. Many asked questions and were convinced of the merits of such a unified framework. I had a few scientists reaching out to me on the following days asking me to develop visualization tools to complement the pipeline.

2.6.3 DATA61 Live Event

DATA61 Live is an event held annually to showcase the science and technology innovations of DATA61 from all over Australia. In 2018, it was held in Brisbane, in our city. The theme was: 'Adapting to Disruption'. We signed up as volunteers and apart from volunteering, we had a chance to attend many lectures, talks and forums. It was a great experience.

Fig. 2.15. DATA61 LIVE Event

3 Conclusion

I worked as a research intern at DATA61 for 24 weeks. During this time, I worked three intertwined projects under the supervision of Nicolas Hudson and Dr. Navinda Kottege.

The first project was developing indoor-Trailnet: a classification based approach to the autonomous navigation problem. For this, first I built a robot with Uvindu Perera for data collection and testing. We designed the power distribution system of the robot and assembled a high level and low level controllers. We calculated the torque requirements for the motors, current, voltage and power limitations for the power supply components and motor controllers during this process. We learnt a lot through debugging the errors we came across and extensively troubleshooting whenever a component or circuit board is damaged to provide a report on the event. I also debugged the driver software for the motor controller and modified parts of it to fix certain issues since it was not being maintained anymore.

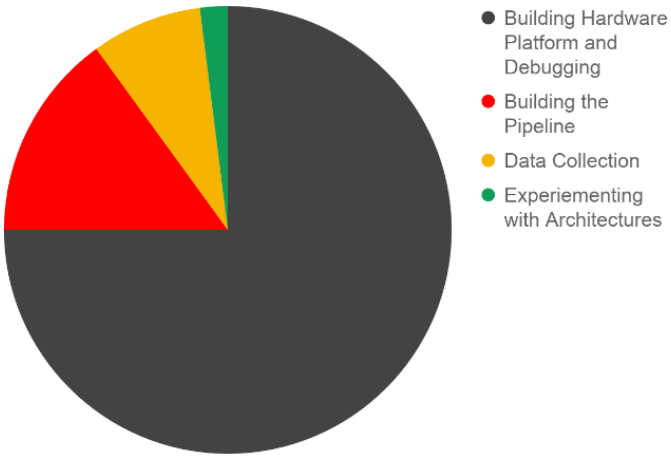

Fig. 3.1. Overview of our Time Spent

I learnt tensorflow and tensorflow-keras for this project and built a 20 layer residual network from scratch and trained it on the supercomputer cluster. Then I learnt tensorRT and deployed the trained network on Jetson TX2: a powerful embedded device that acted as the high level controller for the robot. I also learnt ROS (Robot Operating System) extensively, and created multiple ROS nodes and packages for testing, collecting data from sensors and for controlling the robot.

For the next project of building an end to end pipeline for machine learning in robotics, I experimented and figured out the best practices when training models using large datasets on the supercomputer. I documented these experiments and the results, and proposed an efficient method to set up a writing and reading pipeline. I presented [6] my pipeline in the Robotics Reading Group meeting. Many scientists were interested and some followed up via email asking questions and requesting to build tools for visualizing datasets.

The final project was experimental, where I explored different configurations of neural network architectures to build a robot that can climb hills while avoiding obstacles. Through literature survey and in-depth discussions with my supervisor, I learnt a lot about the structure of Deep Convolutional Neural Networks and possible methods of combining scaler data with the images to train a network. I tried several approaches here: solving the problem as a classification and a regression and using different ways to combine the inputs into the network. However, I was unable to get good results within the time we had there, due to the problems in data collection and the lack of time as we had to spend most of the time building and debugging the robot platform.

Also, I learnt about cross modal learning transfer and neural network distillation process through literature survey as a part of the project that was initially given, before we were changed into a different project a week later. Before and during the internship program, I expressed my interest in working in a project where I can develop algorithms and tackle abstract problems that could lead to new research and a publication. Unfortunately such a project was not available for interns at the time and my supervisors were satisfied with the work I was doing with hardware and deployment of machine learning. However, I learnt a lot through these projects and it had been a great experience. I also got familiarized with the software tools widely used in academia, high end sensors and controllers, and I daily worked with the Bracewell cluster, which is one of the world's largest supercomputer clusters.

In addition to that, I learnt the etiquettes and responsibilities of working as an employee in a company. Helping others and asking for help, attending meetings and following up via official emails, documenting all the tasks and weekly progress in the wiki pages of CSIRO helped me learn a lot about these responsibilities. The work culture in DATA61 is exceptionally inclusive,

where we got to work with people of multiple nationalities and share our culture. The students are allowed to work on their own pace and I was allowed to work overnight on multiple days and work on weekends as well. I also got to attend events such as DATA61 LIVE, where I could attend to many talks and discussion forums and observe the development of cutting edge technology of Australia through the exhibits.

From my experience, I would suggest DATA61 to assess the skills of the interns and assign them to projects relevant to those skills, to utilize their full potential for a project. Also, it would help if they can give an overview of the project at the beginning of the internship and set incremental goals to be completed at given deadlines. I found it disorienting when the project given to me before the internship was changed as I reached there and changed again a week later to settle on an experimental project of my supervisor that subsequently evolved into the three above projects (that I explained in Chapter 2) through the period of six months.

I would also like to suggest NAITA to computerize the supervision process, where interns can submit the intern diary and monthly reports online. This would help because in organizations like CSIRO, the students are expected to maintain an online diary and the interns can save time by writing by hand the same thing they have typed into the online diary.

Therefore, I can conclude that my overall experience in DATA61 was great. I had the opportunity to learn a lot and make contacts. I am deeply thankful to the Industrial Training Division of our university and NAITA for this internship and I am thankful for DATA61 and my supervisors for providing me with such an exceptional opportunity and a training experience.

References

[1] About Paraqum Technologies. https://paraqum.com/about.

[2] About TensorFlow. https://www.tensorflow.org/?hl=hi.

[3] About Wave Computing. https://wavecomp.ai/company.

[4] Llvm. https://en.wikipedia.org/wiki/LLVM.

[5] Mips bought by wave computing. https://www.electronicsweekly.com/news/business/mips-bought-wave-computing-2018-06/.

[6] Presentation: ML Pipeline - DATA61 RRG. https://docs.google.com/presentation/d/1Z7OW8ILDy-wdoWo1Hw84-F0VBjzLAXzpfrpAYbkIddE.

[7] Wave Computing named a top 25 AI solution provider. https://globenewswire.com/news-release/2017/07/20/1054678/0/en/Wave-Computing-Named-a-Top-25-AI-Solution-Provider-for-2017.html.

[8] Wave computing raises 86m usd in oversubscribed series e round. https://wavecomp.ai/wave-computing-raises-86m-in-oversubscribed-series-e-round.

[9] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. *arXiv:1603.02199 [cs]*, Mar. 2016. arXiv: 1603.02199.