SYSTEM DESIGN FOR VISION BASED TRAFFIC SENSING

Project Proposal and Feasibility Report submitted in partial fulfillment of the requirements for the

Degree of Bachelor of Science of Engineering

in

The Department of Electronic & Telecommunication Engineering

University of Moratuwa

Supervisor:

Prof. Rohan Munasinghe
Dept. of Electronic &
Telecommunication Eng.

Group Members:

Abarajithan G.	150001C
Fonseka T.T.	150172A
Wickramasinghe W.M.R.R.	150689N
Wimalasuriya C.	150707V

Co-Supervisor:

Prof. Saman Bandara
Head, Dept. of Civil Eng.

May 2019

Abstract

SYSTEM DESIGN FOR VISION BASED TRAFFIC SENSING

Group Members:
Abarajithan G., Fonseka T.T., Wickramasinghe W.M.R.R., Wimalasuriya C.

Supervisor: Prof. Rohan Munasinghe

Keywords: Traffic, FPGA, Vision, CNN, Hardware Architecture

Through this project, we intend to design and demonstrate a highly optimized hardware architecture that can run a Convolutional Neural Network (CNN) or morphological operations in real time to deduce the level of traffic with acceptable accuracy while being robust to different lighting and weather from the video feed through a camera attached next to the traffic lights. The traffic level sensed through our system can be fed into algorithms developed by traffic engineers (who will be guiding us) to control the traffic lights and make the timing dynamic and sensitive to the traffic level for efficient traffic control. This is a part of the nationwide Intelligent Transport System (ITS) project done in collaboration with the Traffic Engineering Division, Department of Civil Engineering in our university and Road Development Authority (RDA) and funded by the World Bank.

Currently available systems (in developed countries) for vehicle sensing include loop detectors, radar and wireless sensor networks. These methods fail to differentiate between heavy and light traffic conditions and occasionally produce false positives. Existing vision-based systems use outdated technology and they require expensive infrastructure unavailable in a developing country.

Processing the video feed on edge using a CNN on a Field Programmable Gate Device (FPGA) for traffic sensing would be the low cost, scalable solution ideal for a developing country like Sri Lanka. However, it has not been attempted yet, since the compact neural networks and hardware that enable our approach were not available until recently. Hence, we conclude that our approach is unique and of great national importance.

Potential benefits of our project include reduced traffic congestion, saving time and money of millions of people in developing countries. In addition to that, the hardware architecture we intend to develop as our major contribution in this project may be used for other similar applications such as in self driving cars with minimal modification.

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AI	-	Artificial Intelligence
ASIC	-	Application Specific Integrated Circuit
CNN	-	Convolutional Neural Network
ENTC	-	Department of Electronic and Telecommunication Engineering
FPGA	-	Field Programmable Gate Array
IoU	-	Intersection over Union
ITS	-	Intelligent Transportation System
NN	-	Neural Network
OFN	-	Optical Fiber Network
PLC	-	Programmable Logic Controller
RDA	-	Road Development Authority
SoC	-	System on Chip
YOLO	-	You Only Look Once

Chapter 1

INTRODUCTION

1.1 Problem Statement

Traffic lights in Sri Lanka are controlled by static timers with pre-set values. These timers are blind to the dynamic changes in traffic levels. For example, during office hours in the morning and evening, there could be heavy congestion in particular lanes. The static timers do not take the dynamic traffic conditions into consideration and hence they give the pre-set value of green time and red time to all the lanes throughout the day. This causes the congestion to intensify. To solve this, traffic policemen are deployed to critical intersections. They change the pre-set times of the traffic lights and occasionally they turn off the traffic lights and control the traffic themselves.

Engineers from Traffic Engineering Division, Department of Civil Engineering, University of Moratuwa, have developed robust models for dynamic traffic controls using complex algorithms and empirical evaluations using simulation. However, they are unable to implement them since they do not have a low cost, robust and scalable method to sense the traffic level and collect the data for their algorithms.

There are several methods used in developed countries to detect vehicles for traffic light as described in the oncoming sections. However, we find these methods unsuitable for a developing country like Sri Lanka which has heavy, unpredictable traffic. Although vision-based sensing systems are gaining popularity, existing such systems use older algorithms that require more computation power. Hence, they need expensive hardware or infrastructure such as fiber optic cables and coaxial cables laid along the roads, connecting cameras to a centralized system for processing.

1.2 Our Solution

We propose a low cost, distributed system consisting of a camera and a Field Programmable Gate Array (FPGA) that can be deployed near traffic lights that can analyze the video feed and dynamically sense the traffic level. The traffic level information can be further processed by traffic algorithms and a change in time can be proposed to the Programmable Logic Controllers (PLC) that control the traffic lights.

Our major contribution through this project will be a highly optimized hardware architecture that can run a neural network robust to different weather and lighting conditions on a low-cost FPGA deployed next to the traffic lights.

1.3 Objectives

The major objectives of our project are threefold:

1. Collect vision-based traffic level data under various weather and lighting conditions and label them

2. Design morphological operations or choose a suitable Convolutional Neural Network (CNN) and train it through transfer learning to predict the traffic level to an acceptable accuracy, given the video feed

3. Design and demonstrate a highly optimized hardware architecture to implement the above chosen method on an FPGA as a low cost, edge solution

1.4 Uniqueness and National Importance

Through our literature survey, we realized that vision-based traffic sensing has never been attempted on an FPGA before, since the algorithms and hardware were not available until recently. CNNs with low computational complexity and an acceptable accuracy were invented only during the past two years and FPGAs with integrated ARM processors running Linux Operating System (OS) that can dynamically change the hardware configurations in runtime are also new to the market. In addition, since our method is low cost and scalable, without the need of any complex infrastructure, it is a unique and ideal solution for a developing country like Sri Lanka and it could play a major role in solving our traffic problems.

1.5 Potential Applications

Apart from being suitable to solve the traffic congestion problems in developing countries around the world, we believe our hardware architecture with slight modification can be used to run other general mobile CNNs or image processing operations. This opens up the possibility of using our architecture in self driving cars and other similar devices where image processing must be performed in a low-cost chip.

1.6 Navigation to Following Chapters

Chapter 2 of this report analyses and summarizes the findings of our literature survey and examines the existing system in Sri Lanka. Chapter 3 describes the system architecture, possible alternatives, risk analysis, budget, task delegation, timeline and the results of the work we have done so far. Chapter 4 discusses the technical, financial and social feasibility of the project, local and global impact and the feedback from the staff collected during the feasibility presentation and the discussion that followed.

Chapter 2

LITERATURE SURVEY

The following sections provide an understanding of the project background of the key areas such as existing methods for traffic monitoring and control, state-of-the-art CNNs that can be assisted for vehicle detection (which is aimed to be used to estimate the traffic density), FPGA implementation of neural networks and simulation techniques that can be used to test the designed algorithms before implementing in the real-world.

2.1 Existing Methods

Existing solutions for traffic monitoring and control and the pros and cons of each technique are discussed below.

- Static Timers

Static timers are being used for traffic control in most developing countries including Sri Lanka because of its simplicity and inexpensiveness. The most common static timers that are being used are the PLC timers in which traffic lights are operated in a definite sequence. The main disadvantage of this system is the blindness for dynamic traffic level.

- Induction Loops

Inductive loops [13] are coils of wire that are buried on the surface of the road to detect changes in inductance and convey them to the sensor circuitry in order to produce signals. The changes in the traffic signals are caused by the changes in the magnetic field which conveys to the signal controller by the wire coils. This technique is ideal for sparse traffic and not affected by the weather. However, this technique has drawbacks such as difficult for installation and inability to detect small vehicles such as motorbikes which is a regular mode of transportation in Sri Lanka.

- Microwave Radars

These devices can detect large objects travelling towards or away from them. This is done by first transmitting an electromagnetic signal at a constant frequency and measuring the speed of vehicles within its field of view using the Doppler principle, where the difference in frequency between the transmitted and received signals is proportional to the vehicle speed. Even Though this technique is ideal for sparse traffic, this type of detectors cannot detect stationary vehicles which is a major drawback. [13]

- Wireless Sensor Networks

In wireless sensor networks [14] all the vehicles are identified and counted using the Traffic Sensor Nodes (TSNs) and this information is periodically being transmitted to the base station. The aggregated traffic information is then transferred to the traffic signal's time manipulation algorithm to fix the duration of the time for traffic signals dynamically based on the number of vehicles on each traffic signal. Such systems need transmitters and centralized stations for functioning which are expensive.

- Existing vision-based systems

Pole-mounted video detection cameras capture the traffic video feed, connects and transmits the signal for processing hardware via transmission methods such as fibre optics. This technique is not convenient for developing countries such as Sri Lanka, since it requires expensive processing hardware and transmission medium. [9, 3]

- Edge Solutions

In developing countries such as India and Thailand, Raspberry Pi devices are being used as a microcontroller which provides the signal timing based on the traffic density. These devices estimate traffic level using basic image processing techniques such as edge detection and morphological operations. However, this method tends to give false positives due to reasons such as partial occlusions, errors in morphological operations since it does not have a deep understanding about the traffic level. [1, 7, 12]

Traffic controlling techniques such as Induction Loops, Microwave Radars and Wireless Sensor Networks which are used by developed countries are suitable for sparse traffic which is not the common case in Sri Lanka. In most instances we experience unpredictable dense traffic. Thus, we are hoping to give a unique solution as described in the introduction to address the traffic control mechanism of Sri Lanka.

2.2 Neural Networks

State of the art Convolutional Neural Networks such as “You Only Look Once” (YOLO) and SqueezeDet have already come up with optimum object detection algorithms which can be modified in a way that it suits the traffic level estimation task. It is important that the selected model comprises simple computational operations, small in size and function efficiently in order to deploy in an embedded system.

- YOLO v2

YOLO v2 [11] is a single shot real-time object detection system which consist of a simple network architecture which can be assisted for traffic level estimation with a reasonable accuracy. YOLO v2 uses a custom deep architecture darknet-19, a 19-layer network supplemented with 11 more layers for object detection. YOLO makes predictions with a single network evaluation which makes it extremely fast.

This network, which is applied for the input image, divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

YOLO v2 accuracy has been improved compared to the first version by introducing techniques such as adding batch normalization for convolutional layers and using anchor boxes to predict bounding boxes. K-means clustering technique can be assisted to figure out the anchor boxes which have the best coverage for the training data. Since we are dealing with boundary boxes, we use Intersection over Union (IoU) to measure datapoint distances.

- SqueezeDet

SqueezeDet [17] is a single shot fully convolutional neural network inspired by YOLO, which is extremely fast, accurate, energy efficient and small in model size. Here, the stacked convolutional filters extract a high dimensional, low resolution feature map for the input image, which is taken as the input for the ConvDet convolutional layer to compute large amount of object bounding boxes and predict their categories. Finally, these predictions are filtered to obtain the final detections as illustrated in Fig 2.1.

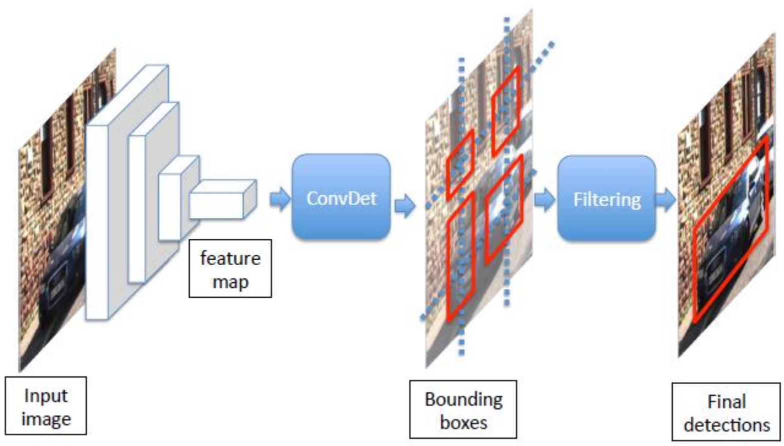

Fig 2.1 SqueezeDet Pipeline

The backbone of this network is SqueezeNet [18] which achieves AlexNet level imageNet accuracy with an extremely small network size. This SqueezeNet is enriched with fire modules which comprised of squeeze convolution layer (which has only 1x1 filters), feeding into an expand layer that has a mix of 1x1 and 3x3 convolution filters.

There are several trade-offs between YOLO and SqueezeDet. For example, YOLO comprises simple architecture with simple computations compared to SqueezeDet, while SqueezeDet is more computationally expensive compared to YOLO. Moreover, SqueezeDet weights file size is small and can be further compressed which is not possible with YOLO. Thus, we must make a wise decision on selecting the most appropriate CNN.

2.3 FPGA Implementation

Commercially available GPU based CNN accelerators for edge processing, such as NVIDIA Jetson TX2[22], Google Coral and Intel Neural Stick[21] are built optimized for a wide range of neural network architectures. They (1-2 GHz) beat an FPGA (0.6 GHz) in frequency but might take many times the clock cycles to run a given neural network when compared to a hardware architecture specifically tailored for the given neural network [20]. Due to this companies such as Microsoft are considering FPGAs for Artificial Intelligence (AI) [4]. Since floating point operations use more resources and are notoriously difficult to be fully pipelined, we explored the possibilities of fixed-point representations and quantized floats [19]. We also found some versions of YOLO and SqueezeDet networks implemented using 8-bit integers without much loss in accuracy [23].

2.4 Simulation

Simulation is an important aspect in our project since it is essential to test our algorithm before implementing on the real-world. For our project, we get the assistance of the Department of Civil Engineering in the University of Moratuwa to interact with the PTV VISSIM software. [8]

- PTV VISSIM

VISSIM is an industry grade micro traffic simulation software used by professionals in different aspects of transport Engineering. It can provide a very accurate simulation of the traffic in a given intersection with properly set parameters. Since the software is widely used in Sri Lanka, the parameters for the Sri Lankan traffic is already available and the software is ready to model the traffic of a desired location with minimal tweaking. Once properly set up, it provides a 3D simulation of the traffic which can be captured by virtual cameras within the software itself and can be exported to a video file.

The algorithm we design cannot be directly applied and tested on the real world. Hence, we use this software to simulate a chosen real-world junction in Colombo with parameters obtained from the Department of Civil Engineering in the University of Moratuwa. These parameters will make the traffic flow in the simulation similar to that of the real-world.

Since the traffic light system in the simulation can be controlled and the flow of traffic accordingly, we plan to extract the real-time traffic data of the simulation and process it with the algorithm predicting the Δt value. This value then will be fed back to the simulation tool in order to change the timing of the traffic lights hence the flow of traffic making a closed loop traffic simulation. This is illustrated in Fig 2.2.

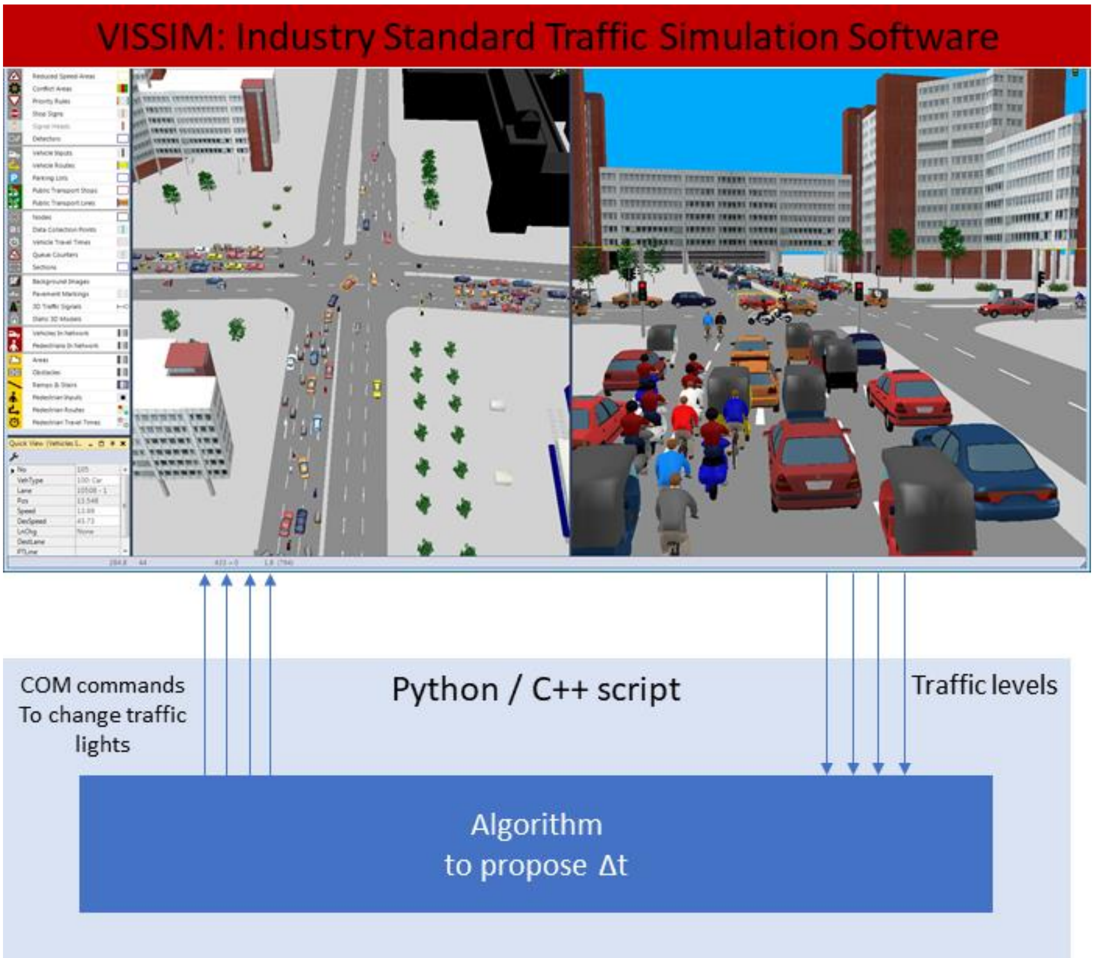

Fig 2.2 Traffic Simulation with VISSIM

Chapter 3

METHODOLOGY

3.1 Scope

- Data collection

In order to train the convolutional neural network, we have planned to obtain data set which is to be collected from 2 to 3 suitable junctions using the data collection device we built. This data set will contain different environmental conditions (day, night, twilight, rain etc.) and this will be labeled appropriately.

To collect the data set, we have built a data collection device with a raspberry pi 3 and a pi camera module. The device is waterproof and can be attached to the horizontal bar of the traffic light pole. The device consists of 2 servo motors arranged using a servo mount in order to obtain the necessary orientation of the camera. The raspberry pi can be powered up at the time of data collection using a power bank at the bottom of the traffic light pole. The raspberry pi can be accessed remotely through Wi-Fi and can be controlled using python scripts written to control the servos and collect data. Moreover, the GUI we made for this device make the data collection efficient and easy.

- Neural Network Design

We will use an existing convolutional neural network which can detect objects and modify/adjust its last layers to suit our requirements in order to detect the vehicles in the road and obtain the traffic level of the road. This chosen neural network will be fine-tuned using the dataset we created and then modified in a way that it predicts the traffic level.

- Hardware implementation

Since we are focusing on a specific Convolutional Neural Network (CNN), we can design a highly optimized hardware architecture for implementing it in the FPGA device which uses less hardware resources.

- Algorithm

We will design an algorithm that takes in the traffic levels in all lanes of the junction from the neural network and throws out a Δt value which will act on top of the existing static timing of the traffic light system. For this project, we limit the scope to four-way intersections only. If the confidence of the sensed traffic level is below a certain threshold, the algorithm will revert to static timing ($\Delta t = 0$). This algorithm will be tested and demonstrated in VISSIM simulation tool.

3.2 Neural Network Architecture and Algorithm

The first step of our approach is to apply simple image processing techniques such as edge detection and morphological operations to estimate the traffic level since this can be easily implemented on FPGA. If an acceptable accuracy was not obtained, we will be moving on to convolutional neural network implementation with an existing state-of-the-art design such as YOLO V2 or SqueezeDet which are more reliable.

The weight set of this network will be adjusted using "transfer learning" since we lack the resources for retraining the model from scratch. This network will be robust to different environmental conditions including day, night, twilight and rain.

After sensing the traffic level, we need to find an algorithm which can predict an appropriate change in static timing (Δt) of the traffic light system. We will be designing an algorithm which is suitable for traffic controlling in Sri Lanka.

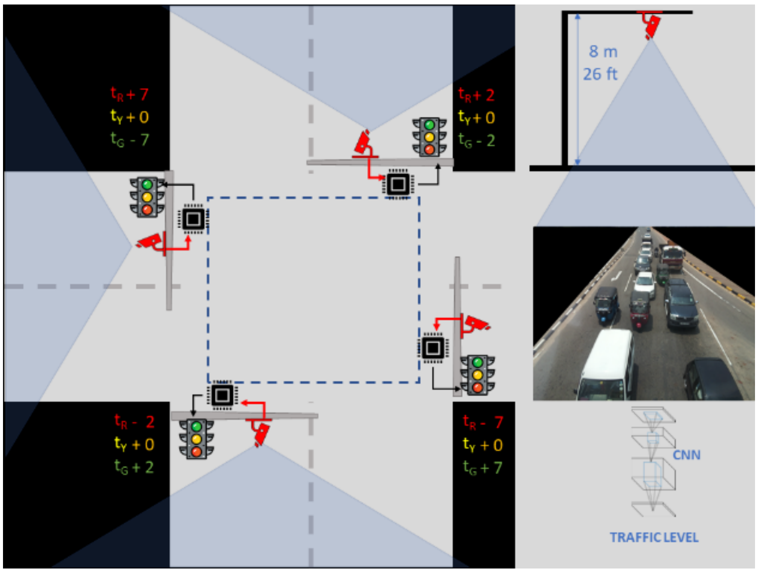

Fig 3.1 System Arrangement

As illustrated in Fig 3.1 the cameras attached to the traffic light pole will collect images of the lanes and send to the processing system implemented on FPGA (described in next section), where it will run through the neural network and the mentioned algorithm and communicate with the adjoined lanes' devices to come up with Δt which will act upon the static timing (shown as t_R, t_Y, t_G) for efficient traffic flow.

3.3 Hardware Architecture

The main objective of the project is to give an edge solution to the traffic problem. Therefore, we plan to implement the chosen neural network in a FPGA device. FPGA implementation of a CNN can be a complex task due to the resource constraints in the FPGA. For this, we need to design a specific hardware architecture for the CNN that contain modules such as convolutional units, adder trees, multipliers etc. using Verilog while optimizing them to use less resources as convolution can be highly computationally expensive.

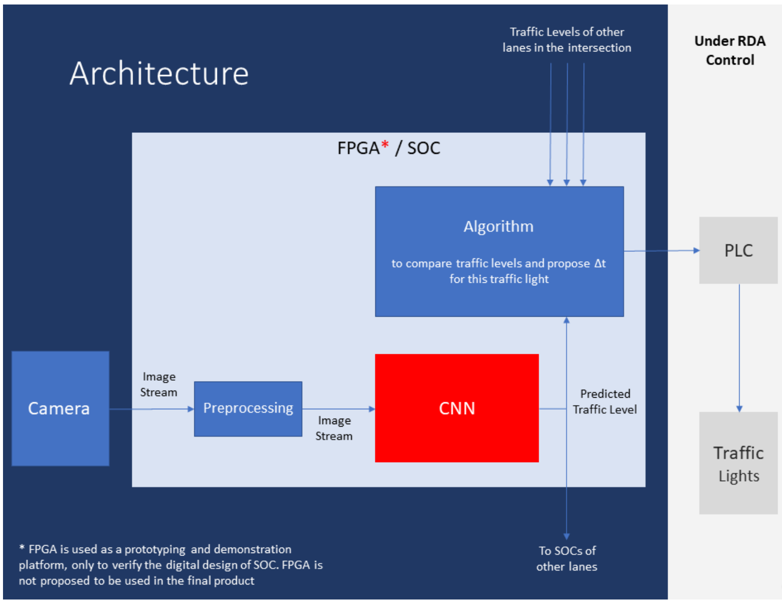

Fig 3.2 Hardware Architecture

3.4 Risk Analysis

Risk Factors of the project and Risk Management techniques are summarized in Fig 3.3

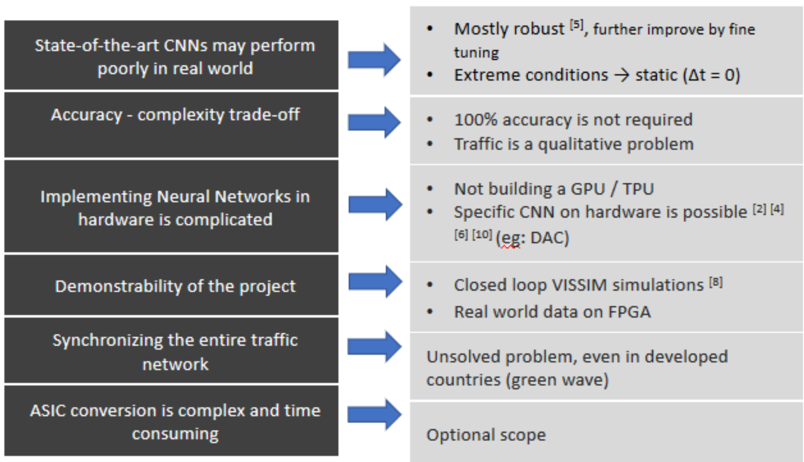

Fig 3.3 Risk Factors

3.5 Budget, Task Delegation and Timeline

Table 3.1 Budget Proposal

	Amount (Rs.)
Raspberry Pi 3 Model B (x2)	14,000
Pi Camera (x2)	4,000
FPGA Board (x4)	36,500
FPGA Camera (x4)	14,000
GPU Server (Estimated GCP computational cost)	25,000
Material to build the data collection device	10,000
Total Estimated Amount	**103,500**

Table 3.2 Task Delegation

Task	Abarajithan	Tehara	Rukshan	Chinthana
Literature review & analyzing alternate methods				
Building & testing data collection device				
Implementing device and collect data				
Compare different approaches				
Modify a suitable CNN				
Collect data from 3 junctions & train				
Implement convolution blocks in FPGA				
Design hardware architecture for CNN				
Implement hardware architecture for CNN				
Test on real world data				
Train & test same CNN in simulation data				
Design algorithm to predict Δt and test in simulation				

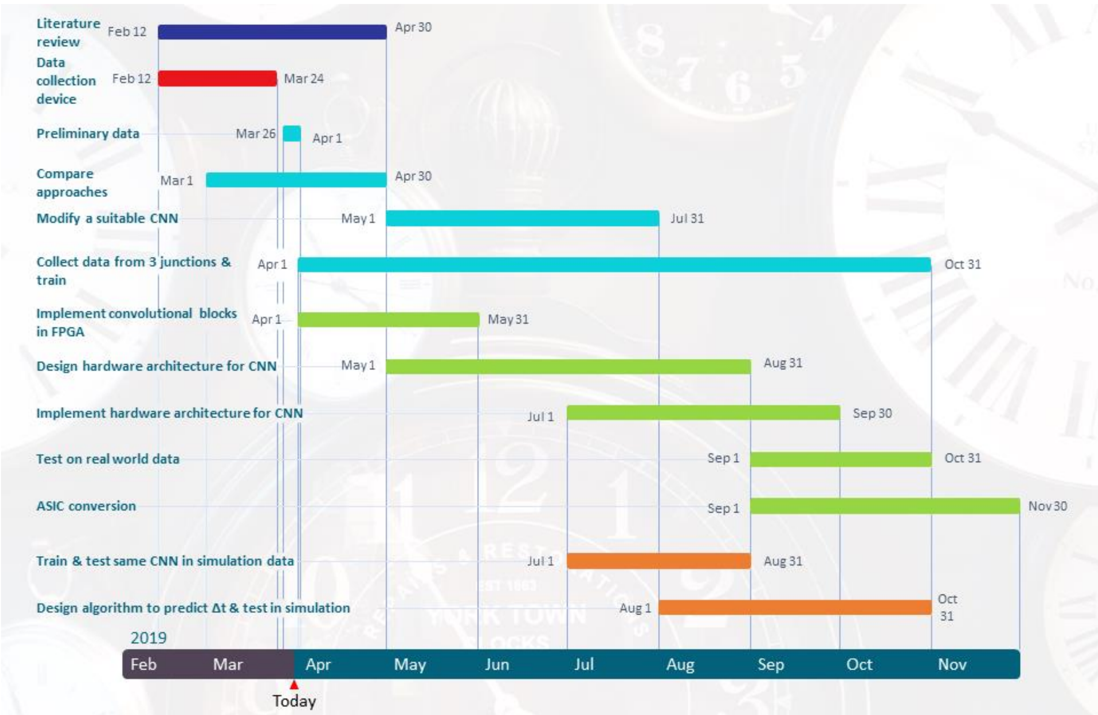

Fig 3.4 Project Timeline

3.6 Initial Results

Fig 3.5 Data Collection Device and Setting Up

In order to collect preliminary data, we were able to fix the data collection device we built with the help of a bucket crane vehicle as shown in Fig 3.5.

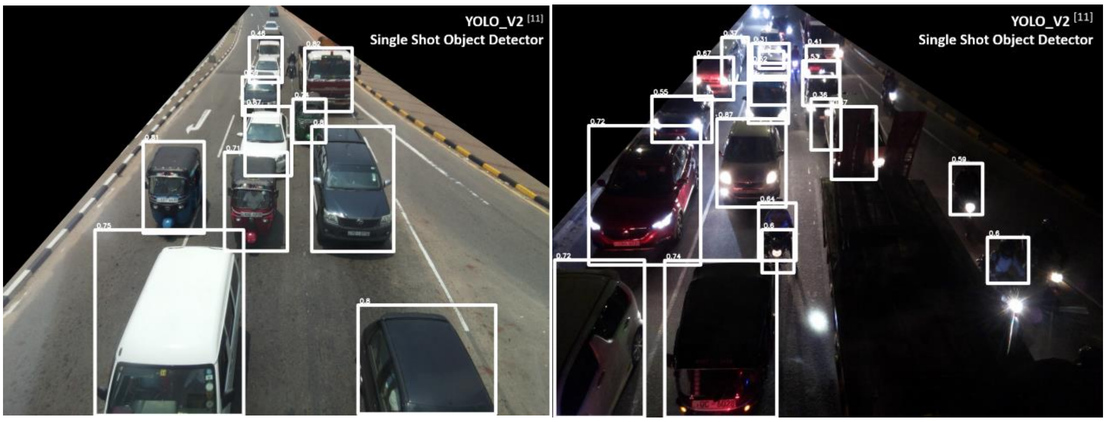

Fig 3.6 YOLO V2 Detection on Daytime/Night-time

Fig 3.6 illustrates some of the initial results we got for the preliminary data after running detections using YOLO v2. These results can be further optimized using fine-tuning with the dataset that is planned to create. These optimized detections will then be used to determine the traffic level so that it can be fed into the algorithm to predict an appropriate time change.

Chapter 4

DISCUSSION AND CONCLUSION

4.1 Findings of the Literature Survey

The literature survey revealed that there are numerous ways to sense traffic, but they have their own downsides and no method is perfect for the job. Mostly it is a matter of accuracy and sensitivity. Most of the time they are very expensive or require advanced infrastructure such as Optical Fiber Network (OFN) linkages and long-range networking to be already in place. Some methods such as induction loops and pressure plates require literally digging the road out to install the system. This project addresses all these problems by being very accurate, relatively cheap to install and requiring no external infrastructure to work except for standard traffic light systems, which are very commonplace.

Speaking of similar solutions, there has been no attempt to use dedicated System On Chips (SoCs) for this application, meaning we are likely in unexplored waters, but all the required technologies already exist and are used widely enough to consider using them safe. The approach that is closest to ours on this problem is the usage of raspberry pi systems to analyze vision traffic but usage of object detectors on dedicated SoCs can outperform simple solutions like morphological operations or edge detection running on weak hardware such as raspberry pi, in terms speed, accuracy and reliability.

The neural network to be used in the final solution is not fully decided yet but it is likely to be one of the embedded capable state-of-art networks, YOLO or SqueezeDet. These networks are suitable to be put on FPGA/SoC platforms and operate at adequate speeds. Exact performance numbers can only be obtained through hands on testing, but they certainly are fast and accurate enough to be viable for the proposed solution.

4.2 Technical Feasibility of the Project

Speaking of the technical feasibility of the project, it exploits a range of cutting edge yet widely used technologies. Object detection networks are becoming more and more commonplace, appearing in products like drones and mobile phone cameras. It has been proven in our initial data collection that these networks can very effectively detect vehicle traffic. SoCs have already burrowed their way deep into our lives, everywhere from mobile phones to wrist watches. Putting an embedded neural network on a SoC is not very commonplace but not an untested area either, since competitions like Design Automation Contest have received successful implementations of this technology. PLC technology currently used to control the traffic lights is relatively simple and since this project is backed by RDA, interfacing with the existing traffic light infrastructure should not be a problem. Thus, we can safely assume that putting an object detection network inside a SoC to detect and control traffic is a feasible and practical idea, technology wise.

During the feasibility presentation, we were questioned Dr. Prathapasinghe and Dr. Premarathna how we can solve a seemingly global optimization problem like traffic by optimizing individual junctions and how we would handle synchronization between junctions. The question was answered by Dr. Jayasooriya from civil engineering department, stating that they already have very efficient local optimization algorithms to solve the problem but without a proper way to sense the traffic level at intersections, they have had little success deploying the systems. It was then established by him that what they want from this project is an accurate way of measuring the traffic levels and then they had means of effectively controlling that traffic. This was then further clarified by Dr. Dayananda, the head of department of ENTC, mentioning that this project will be evaluated only on the electronic aspect, which is the sensing part and not based on how the traffic flows, hence eliminating optimization and synchronization from the scope of this project. Dr. Prathapasinghe also agreed that although these CNNs exist, designing a custom architecture to accelerate them and implementing it on FPGA is a task complex enough as our contribution in a final year project.

4.3 Financial Feasibility of the Project

This project is part of the ITS and is backed by the world bank. They are ready to invest in the goal of building a safe and efficient transportation system, securing the financial side of the project. However, the expenses for the project is relatively limited since we will be using low cost, off the shelf hardware and educational software. We initially planned to design an ASIC out of our architecture, which would be quite expensive. However, during the feasibility presentation, Prof. Jayasinghe and Dr. Pasqual commented that ASIC manufacturing is very expensive, even when mass manufacturing this device to meet the demands of a small country such as Sri Lanka. Hence, they suggested to use an FPGA itself in each device, hence lowering the cost and improving design flexibility and enabling updates to architecture over time.

4.4 Social Feasibility of the Project

Even among the normal people, practically everyone is aware of and most likely has faced the hardships of traffic jams and inefficient traffic management. All of them will gladly accept the installation of a system that clears the traffic from intersections better and allows them to get on their way faster. Nobody likes to wait at a red light closing a more congested lane and opened up to a less congested lane just because the traffic light is too dumb to understand which side actually has traffic. If very early testing was carried out at a live intersection, it may disturb the lives of people, which is the precise reason that the preliminary tests are done on the simulation. Only once the system is fully ready it will be tested at a live intersection and at that point, system will be ready to handle the real-world conditions and it should cause only a positive effect on the traffic flow. After testing, if the system is properly implemented without any unexpected faults, it should work in everyone's benefit and no one should have any negative concern towards the existence of the system, proving it is socially feasible too.

4.5 Local Impact of the Project

Traffic management is one of the major problems of the modern world, even in developed countries. Sri Lanka is no exception in this case. This project, when applied over the existing static timing system of traffic control systems, can dynamically allocate the road resources to the otherwise unpredictable vehicle traffic, ultimately leading to a better transport system. This will be helpful to a very large number of road users in their day to day lives. In a developing country like Sri Lanka, an intelligent traffic control system will be a significant achievement.

4.6 Global Impact of the Project

This project can be a steppingstone to several other projects. This can, in a future iteration, be integrated with an intelligent pedestrian crossing system, allowing to cut further down on saving time in the intersection. Multiple junctions may be able to talk to each other, routing traffic more efficiently, and at one point, it might be the key to unlocking the green wave, which is a proposed system that is being tested in developed countries, which would allow a vehicle to pass through a number of junctions uninterrupted once it is caught in a group of vehicles let in through one traffic light, but has not been successfully implemented anywhere yet. Another possible development is to interface this with V2X communication protocols, which can aid the system to place specific vehicles within the view of the system and coordinate more efficiently. Finally, the object detector may be modified to identify special vehicle types such as ambulances and fire trucks and let them through faster than regular traffic. All these are possible future developments, but the system will be developed in a way that these can be put together rather easily.

4.8 Conclusion

Looking at the discussion of feasibility and significance, one clear fact stands out. We have the capability to build a powerful tool with a massive significance, which is feasible within the discussed context. There is a proof of concept analyzed dataset depicting the robustness of the technology. Neural network acceleration on FPGA/SoC has been already done in various other fields. There is an industry approved testing ground to test the system on (Vissim simulator). And most importantly, the project is backed in one way by the RDA and by the world bank the other way. At the end of feasibility presentation, the ENTC staff, a group of highly experienced and reputed academics agrees that the project is feasible in all aspects and complex enough as a final year project, and it is an important contribution to the society and the field of engineering while commending the initial work we have done in a short time. So, we arrive at our conclusion which is, vision-based traffic sensing and control is a highly feasible and a globally and locally significant project to be chosen and carried out as a final year project of the department of Electronic and telecommunication engineering.

REFERENCES

[1] K. Vidhya and A. B. Banu, "*Density Based Traffic Signal System*," International Journal of Innovative Research in Science, Engineering and Technology, vol. 3, no. 3, Jan. 1970. http://www.rroij.com/peer-reviewed/density-based-traffic-signal-system-50686.html

[2] K. Mohammad and S. Agaian, "*Efficient FPGA implementation of convolution*," in 2009 IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 3478–3483. https://ieeexplore.ieee.org/ielx5/5340904/5345886/05346737.pdf?tp=&arnumber=5346737&isnumber=5345886

[3] P. Duse, A. G. Gaikwad, and A. Chhajalane, "*Intelligent Traffic Control System for Congestion Control and Violation Detection at Traffic Signal*," Imperial Journal of Interdisciplinary Research, vol. 3, no. 5, May 2017. www.imperialjournals.com/index.php/IJIR/article/download/4894/4704

[4] "*Microsoft's Project Brainwave brings fast-chip smarts to AI at Build conference - CNET*." https://www.cnet.com/news/microsoft-project-brainwave-speeds-ai-with-fpga-chips-on-azure-build-conference/

[5] "MIT Traffic Data Set." http://www.ee.cuhk.edu.hk/~xgwang/MITtraffic.html

[6] R. Zhao, X. Niu, Y. Wu, W. Luk, and Q. Liu, "*Optimizing CNN-Based Object Detection Algorithms on Embedded FPGA Platforms*," in Applied Reconfigurable Computing, 2017, pp. 255–267. https://link.springer.com/chapter/10.1007/978-3-319-56258-2_22

[7] K. Kiratiratanapruk and S. Siddhichai, "*Practical application for vision-based traffic monitoring system*," in 2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009, vol. 02, pp. 1138–1141. https://ieeexplore.ieee.org/ielx5/5076159/5137082/05137245.pdf?tp=&arnumber=5137245&isnumber=5137082

[8] PTV Group Traffic, *PTV Vissim*: Simulation of a Complex Intersection. https://www.youtube.com/watch?v=OtYby7QnyAE

[9] "*The use of fiber optics in security and surveillance systems - Cabling Install*." https://www.cablinginstall.com/articles/print/volume-19/issue-3/features/the-use-of-fiber-optics-in-security-and-surveillance-systems.html

[10] P. V. F. U. States, "*Why CTOs Should Reconsider FPGAs*," Datanami, 26-Mar-2019. https://www.datanami.com/2019/03/26/why-ctos-should-reconsider-fpgas/

[11] J. Redmon and A. Farhadi, "*YOLO9000: Better, Faster, Stronger*," arXiv:1612.08242 [cs], Dec. 2016.
www.arxiv.org/pdf/1612.08242.pdf

[12] Y.-C. Chung, J.-M. Wang, and S.-W. Chen, "*A Vision-Based Traffic Light Detection System at Intersections*," A Vision, p. 20.

[13] "*How Traffic Lights Sensors Work | Automate Systems*," Auto Mate Systems, 20-Oct-2015. http://www.automatesystems.co.uk/how-traffic-light-sensors-work/.

[14] "*Traffic Light Sensor Technology*," AZoSensors.com, 27-Nov-2012.
https://www.azosensors.com/article.aspx?ArticleID=95

[15] M. S. Bains, B. Ponnu, and S. S. Arkatkar, "*Modeling of Traffic Flow on Indian Expressways using Simulation Technique*," Procedia - Social and Behavioral Sciences, vol. 43, pp. 475–493, 2012. https://pdf.sciencedirectassets.com/277811/1-s2.0-S1877042812X00144/1-s2.0-S1877042812010026/main.pdf

[16] "*Quantizing Neural Networks to 8-bit Using TensorFlow Lite*." ARM Limited, 2019.
https://developer.arm.com/solutions/machine-learning-on-arm/developer-material/how-to-guides/quantizing-neural-networks-to-8-bit-using-tensorflow

[17] B. Wu, A. Wan, F. Iandola, P. H. Jin, and K. Keutzer, "*SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving*," arXiv:1612.01051 [cs], Dec. 2016.

http://arxiv.org/abs/1612.01051

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "*SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*," arXiv:1602.07360 [cs], Feb. 2016.

http://arxiv.org/abs/1602.07360

[19] J. L. Gustafson and I. Yonemoto, “Beating Floating Point at its Own Game: Posit Arithmetic,” p. 16.

[20] K. Mohammad and S. Agaian, “Efficient FPGA implementation of convolution,” in *2009 IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, 2009, pp. 3478–3483.

[21] “Get Started | Intel® Neural Compute Stick,” 07:00:00 UTC. [Online]. Available: https://software.intel.com/en-us/neural-compute-stick/get-started.

[22] “Jetson TX2 Module,” *NVIDIA Developer*, 01-May-2017. [Online]. Available: https://developer.nvidia.com/embedded/buy/jetson-tx2. [Accessed: 10-Apr-2019].

[23] K.O, *Low Precision High Performance for darknet. Contribute to k5iogura/darknet_lph development by creating an account on GitHub*. 2018. https://github.com/k5iogura/darknet_lph