Low Cost Machine Learning Platform Development and its Application in Self-Driving

Abstract

The purpose of this project is to find a way of turning widely used general purpose robotics kits, such as those used for FIRST Tech Challenge (FTC), into a machine learning platform.

To complete this conversion, some important hardware and software changes were made on existing FTC platform. Key hardware elements to the conversion were the addition of a remote control system, a laptop, and a webcam. Software elements necessary for this conversion include image and neural network modules. These can be found online and can be used for free.

To make hardware and software components work together, two communication systems between the controller and laptop were proposed. One option was to establish one-way communication between controller and laptop, while the other option was to establish two-way communication between the controller and laptop. The first option was used in this project as it allowed for more flexibility.

To verify that the developed machine learning platform functions as expected, the converted platform was applied to a typical machine learning application: the self-driving. A complete machine learning process including data collection, neural network model training, and actual self-driving was successfully displayed with the developed platform.

The system conversion proposed in this project can be achieved at low cost and can also be easily applied to a variety of general robotics kits.

Introduction

These days, artificial intelligence, deep learning, and machine learning are everywhere [1][2][3]. They are so hot that it has become a revolution across all industries. Two reasons lying behind this immense increase in popularity in recent years: the abundance of data and the increase in computing power. A well-known example of machine learning is its application in self-driving vehicles [4].

Due to this new found popularity, many high school and college students are eager to learn, especially those involved in robotics. There are many high school and college robotics teams and events around the world [5], and many of them are wondering if they can take advantage of the robotics kits they own to learn machine learning through hands on experience.

In this work, a study was performed to explore the possibilities of developing a machine learning platform that is easily accessible to those people and could be done at a very low cost. The goal is to develop a machine learning platform that was based on a general purpose robotics kit, which many teams or students would already have.

That earliest version of laptop robots [6] was likely developed as early as 2002. They were first manufactured as a kit and sold at a very high price. Besides regular robot functions, they had very limited vision capabilities. The low-cost robot mentioned in [7] were first made in 2002 and was designed by a college for vision education and competition purposes. Unlike the robot in [6], this robot has gone through many different evolutions and is still being used by some robotics teams. This robot, however, requires a lot of maintenance and is mainly geared towards vision related robotics activities.

More recently, DIY based robots [8] have been applied for machine learning purposes. Some enthusiasts even meet up on weekends to race their self-driving cars [9]. All these robots are single board computer based instead of laptop. On top of that, these robots require lots of labor intensive work like wiring and soldering. This isn’t ideal for people or teams who are just interested in learning the basics of machine learning.

Latest developed robots which can be used to learn machine learning and deep learning can be found in [10]. This robot uses the latest technology, such as a robot operating system and a graphical processing unit (GPU). Learning this new system is very complicated for beginners, and this kit is very expensive.

The focus of this study is to explore the way to develop a low cost and education-oriented machine learning platform which is based on existing regular robotics kits.

Method

1. General Purpose Robot Unit:

There are many general purpose robot kits available on the market. All of them have many things in common. They have a controller, sensors for input, and motors for movement. It is programmed through software on a standalone laptop and some come with an R/C system as shown in Fig.1.

2. Machine Learning Platform:

(1) Hardware features required by a machine learning platform are: (a) a powerful data processing unit (laptop) (b) remote control for data collection purposes (c) webcam for grabbing images

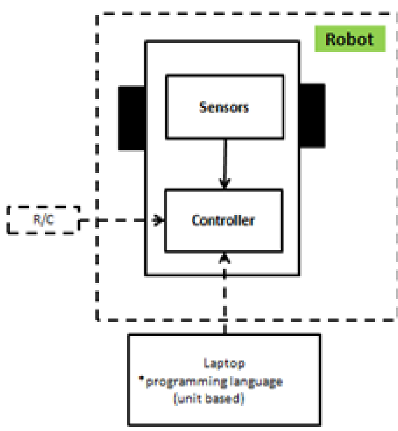


Fig. 1 Diagram of a general-purpose robot unit

(2) The design ideas proposed in this project are shown in Fig. 2. There are two options. Option one seen in Fig 2 (a) has the remote-control system connected to the laptop. This option allows a

robot that didn't originally have a remote-control option to have that function. The second option can be seen in Fig 2 (b) where the remote-control system is connected directly to the controller [12]. This option frees up one USB port on the laptop.

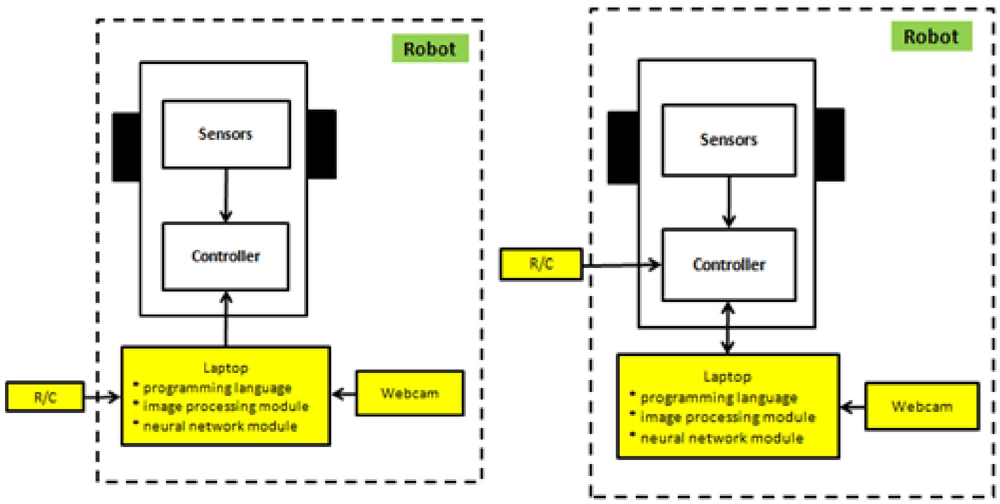


Fig. 2 Design options of a machine learning platform

(a) Option 1 (left): one way communication between laptop and controller

(b) Option 2 (right): two way communication between laptop and controller

(3) Software Environment of Machine Learning Platform:

(a) a machine learning friendly programming language on the control laptop
(b) image processing module [11]: image processing is used to grab the picture, save image array, save label array
(c) neural network module [11]: there are many kinds of open source neural network modules. For this project, a neural network module that comes with the image processing module is used.

(4) System integration:

In this project there are three USB communications, as shown in figure 2

(a) Communication between R/C receiver and robot laptop

(b) Communication between webcam and robot laptop

(c) Communication between controller and robot laptop. This is very important to the functioning of the machine learning platform as labeling during data collection is needed.

(5) Converting the general purpose robot to the machine learning platform doesn't cost much, if at all. Teams would only need to buy a webcam (or use the built in laptop webcam) and a remote control system if need be. This would cost teams with webcams and R/C systems very little.

Results

1. Converted Machine Learning Platform

(1) Fig. 3 and 4 show the result of the conversion from general robot kit to machine learning platform. The basic structure remained the same. All that was done was the top structure was removed so that a laptop could be mounted on the robot.

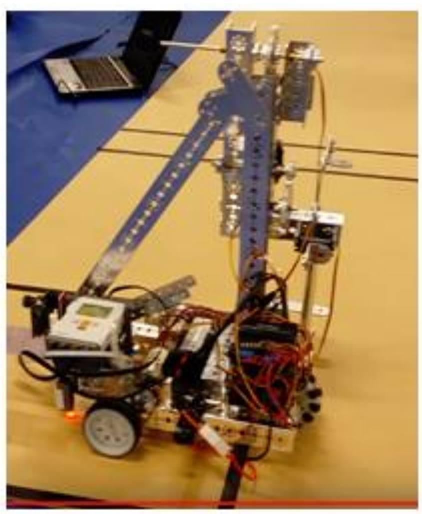


Fig. 3 General Purpose Robot Unit (Top 8th placed bowling robot at WRO Qatar world finals)

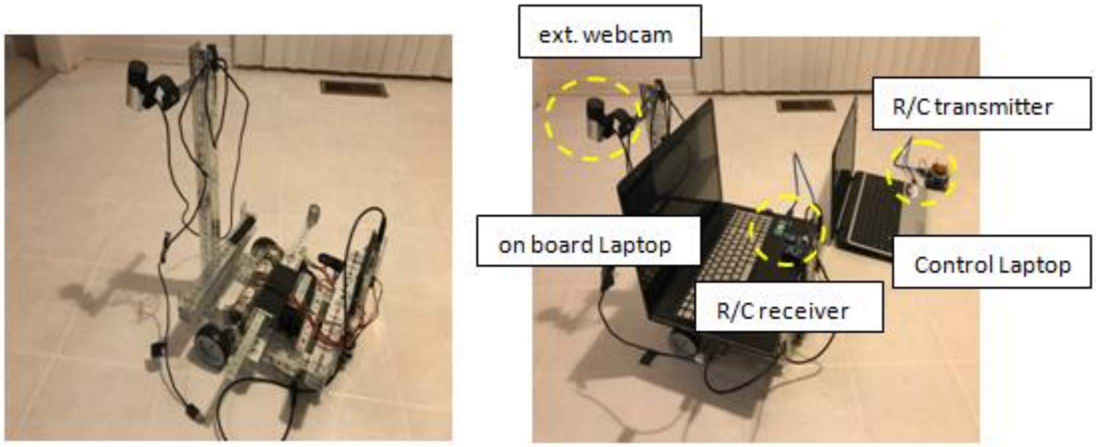


Fig. 4 Converted machine learning platform (EduBot)
(a) platform w/ laptop removed (left)
(b) platform w/ laptop and R/C system(right)

(2) A webcam holder was added so that the robot could have vision capabilities. A built in laptop webcam could also be used if an external webcam is not available.

(3) Remote control system

A key board based R/C system was developed in this project. a second laptop was set up to serve as the control center as seen in Fig. 4, and micro controllers were connected to both laptops. Transmitters were then connected to the micro controllers which allowed for wireless communication between the two laptops.

The cost for the conversion is low, esp., considering the fact that laptops are readily owned by robotics teams. The estimated cost excluding laptops is shown in the table below:

Tetrix Units	PRIZM	Webcam	Keyboard RC (2XArduino+Laptop)	or Joystick RC (1XArduino+JS Module)	Laptop (processing unit)	Total cost excluding laptop
New set	Avaialble	$25.00	$40.00	$29.00	Available	$65.00
Old set	$180.00	$25.00	$40.00	$29.00	Available	$245

2. Machine Learning Platform Application in Self-Driving

This system was applied to self-driving to evaluate if the machine learning platform was functional or not. Three steps are necessary to achieve self-driving, as shown in Fig.5. The first step is data collection, then model training, then finally self-driving (inference).

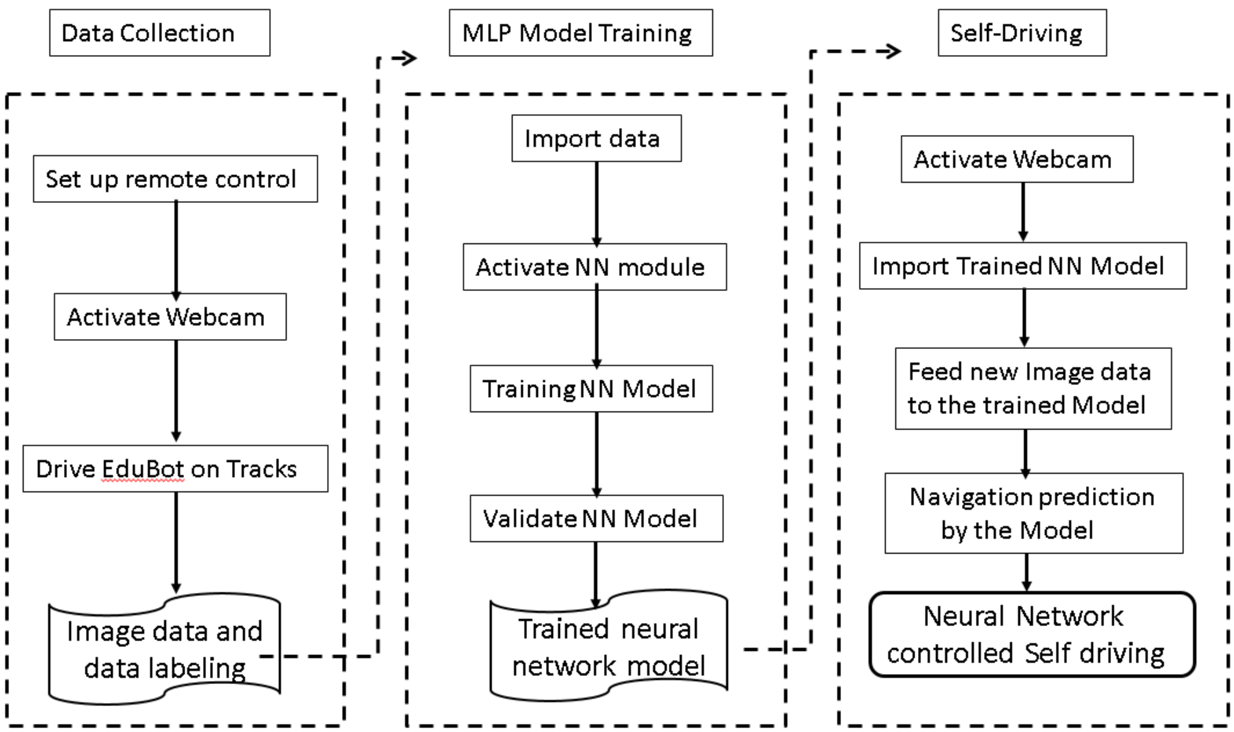


Fig.5 Self-Driving Implementation in Developed Platform (EduBot)

(1) Data Collection

A way of collecting data is needed to have any machine learning application. This was done through the keyboard R/C system. Every time the user pushes a key down the program saves the image the camera is currently seeing as well as the corresponding label, as shown in Fig. 6. The raw data were cleaned up and then processed in the format that could then be used for NN training in OpenCV, as explained in detail in Fig.7.

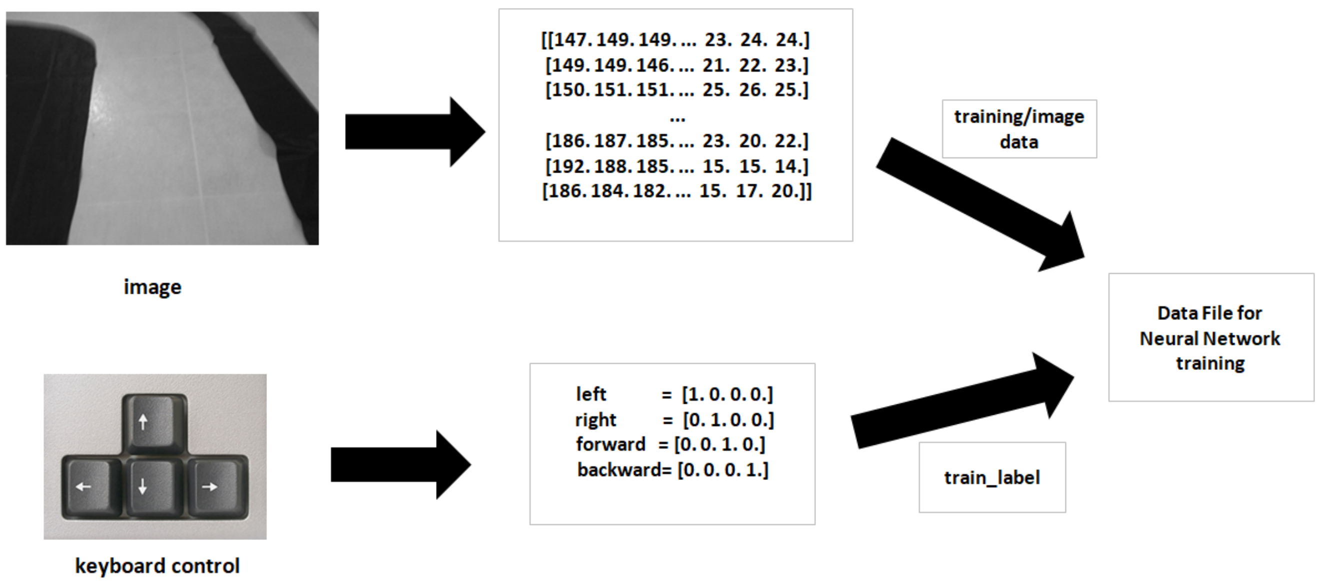


Fig. 6 data collection process

Raw Data Collection	Data Cleaning & Checking	Data manipulation & formatting
•Use R/C and webcam to take images •Raw Image stored in form of Numpy 2D matrix •Black and white image of size 240X320 in this project. •Record keyboard actions as 1D array for data labeling at the same time images is taken.	•Make sure images have same format (size/resolution) •Make sure labels are correct. •Find low quality images and cut them out. •Check if the dataset is well balanced to ensure that no bias is occurs during model training	•Reshape the image 2D Numpy array into a 1D array •The reshaped image is paired with training labels (1D Numpy array) • The paired image data and labels are saved into a Numpy format file, which will be used as input data during model training.

Fig.7 6 Data Collection and Processing

(2) Neural Network Training

Next is model training. MLP or multilayer perception was used as the neural network model [13]. It's the classical kind of neural network. It is comprised of three or more layers. Data is fed into the input layer, goes through one or more hidden layers, and then predictions are made on the output layer, as shown in Fig.8.

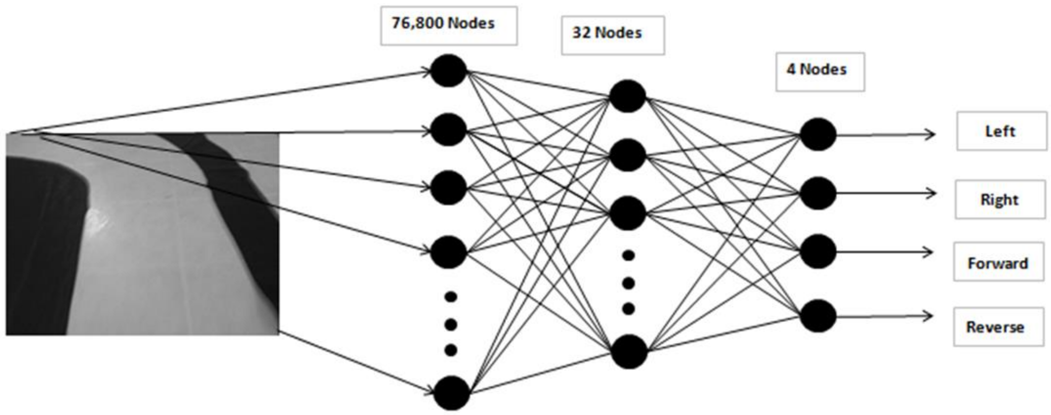


Fig. 8 Neural network model

Multi-Layer Perceptron (MLP) used in this study is a general feedforward network, and it uses backpropagation algorithm to train the model in following two steps: (1) Forward pass phase: computes 'functional signal', feed forward propagation of input pattern signals through network (2) Backward pass phase: computes 'error signal', propagates the error backwards through network starting at output units (where the error is the difference between actual and desired output values)

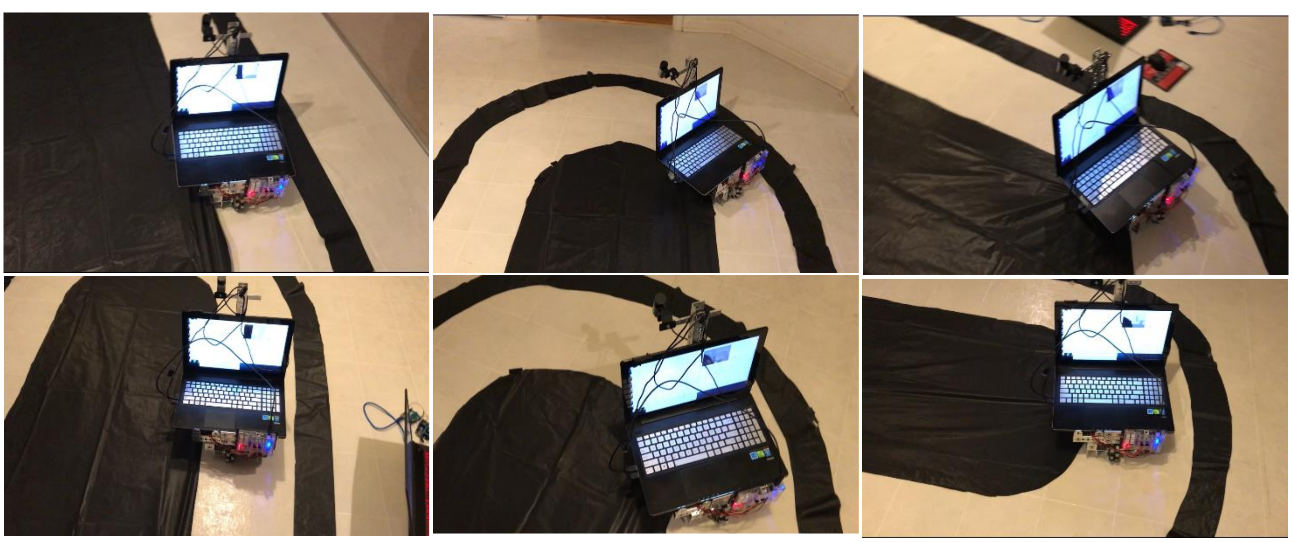


Fig.9 The Developed Machine Learning in Action: Self-driving on a Test Field

(3) Self-Driving

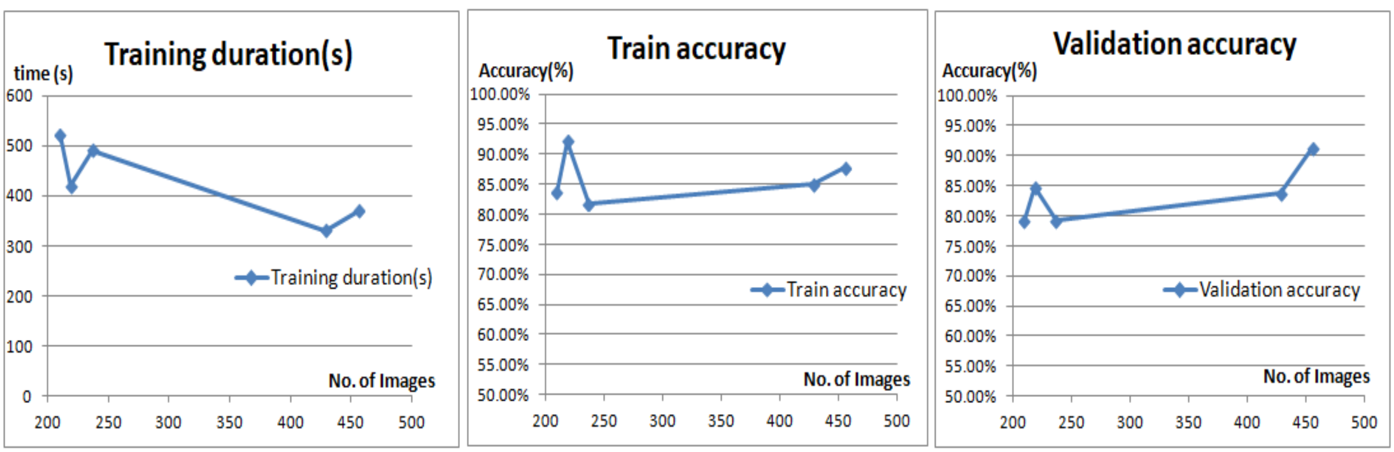
Fig 9 shows some photos taken from Edubot self-driving operation at a few key locations on a test track. The EduBot is accomplishing self-driving using the trained MLP. It is moving around the track through the use of the trained model. The program uses the new images seen by the camera as input for the model, which then outputs the predictions as to where the robot should move.

3. Platform Performance Evaluation

(1) The laptop used for NN training was not very advanced, and its main specs were:

Processor: CPU @1.80 GHz

RAM: 16.00 GB



(a) (b) (c)

Figure 10: Performance study in neural network training: (a) Left: image numbers vs. training time (b) Middle: image numbers vs. training accuracy (c) Right: image numbers vs. validation accuracy

(2) Performance study on the model training

As shown in Fig.10, even with a relatively low performace laptop, the system could collect up to 500 images and finish the training within a few minutes, making it a very good platform for performing machine learning related activities.

Conclusions

1. This project has established, for the first time ever, communication between Python and PRIZM. Based on this communication, a machine learning platform was developed from a Tetrix based unit.
2. In this project, two design options were proposed. One option was to establish one way communication between controller and laptop. The second option was to establish two-way communication between the controller and laptop. The first option was used in this project as it allowed for more flexibility.
3. The key hardware elements to the change from general purpose robotics kit to machine learning platform were the addition of a remote-control system, a laptop, and a webcam. These are only slight changes on an existing robotics unit and are very easy to do.
4. There were also software elements necessary for this conversion including image and neural network modules. All these modules are open source and can be used for free.
5. Proposed machine learning platform was applied to self-driving by successfully going through data collection, model training, and finally self-driving (inferences).
6. The performance study of the developed machine learning platform shows that the platform can be used to collect enough data and perform model training for education-oriented machine learning activities, such as the self-driving accomplished in this project, without using a super powerful computer.
7. The platform change proposed in this project can be done at low cost and be easily applied to a variety of general-purpose robotics kits. This has provided a great help for robotics teams or people who are eager to learn machine learning through robotics.

References

[1] Richard Socher, "Where AI is today and where it's going:" (TEDx Talks), https://www.youtube.com/watch?v=8cmx7V4oIR8&t=56s
[2] Sundar Pichai, "How machine learning & deep learning improved technologies", https://www.youtube.com/watch?v=czVeSFH4dWc&t=53s

[3] TOBY WALSH, “The AI Revolution”, https://education.nsw.gov.au/media/exar/The_AI_Revolution_TobyWalsh.pdf
[4] Chris Urmson,” How a driverless car sees the road” (Ted Talk), https://www.youtube.com/watch?v=tiwVMrTLUWg
[5] Wikipedia, “Robot Competition”, https://en.wikipedia.org/wiki/Robot_competition
[6] Bob Sullivan, “a robot or a laptop on wheels?”, http://www.nbcnews.com/id/3078544/ns/technology_and_science-tech_and_gadgets/t/robot-or-laptop-wheels/#.XA1de3RKiM8 (ER1)
[7] CJ.Chung, “introduction to L2Bot”, https://www.robofest.net/l2bot/introL2Bot.ppt
[8] Zheng Wang, “Self Driving RC Car”, https://zhengludwig.wordpress.com/projects/self-driving-rc-car
[9] Harriet Taylor, “These tech CEOs and Googlers spend weekends racing self-driving cars at a warehouse in Oakland”, https://www.cnbc.com/2017/04/12/tech-ceos-and-googlers-self-driving-car-races.html
[10] Nvidia, “Go Hands-on with 'Jet' - A DIY Robotics Kit”, https://developer.nvidia.com/embedded/learn/jet-diy-robotics-kit
[11] Open Source Computer Vision, “OpenCV-Python Tutorials “, https: //docs.opencv.org/master/d6/d00/tutorial_py_root.html
[12] PITSCO, “Tetrix-prizm-programming-guide-44716.pdf “, https://asset.pitsco.com/sharedimages/resources/tetrix-prizm-programming-guide-44716.pdf
[13] Jason Brownlee, “Crash Course On Multi-Layer Perceptron Neural Networks”, https://machinelearningmastery.com/neural-networks-crash-course/

Acknowledgment

I would like to thank PITSCO for sponsoring the work in this project.
I would like to thank LTU Professor CJ Chung for helping me over the course of this project and providing needed resources.