Jim Langfeldt & Kyle Marsh

AI/Robot Vision in FRC Robotics

Penn KINGSMEN

FIRST ROBOTICS COMPETITION

Jim Langfeldt
jlangfeldt@phm.k12.in.us
574-254-2881
@JimLangfeldt

Kyle Marsh
marshk@mishawaka.k12.in.us
574-254-7300
@marsh_mhs

Courses impacted: Robotics Classes/Teams: Our students working during the school day and many hours outside of school to design, build, and compete with a robot. Many rules and specifications must be met from FIRST. The class is divided up into multiple sub teams working together to complete the robot(s). Students will specialize in certain areas. “Teachers” are mostly serving as facilitators.

The Robotics Classes will be heavily impacted from our work with RET. We have developed a unit around AI/Vision Tracking. These areas continue to be a major emphasis in robotics and in our competition.

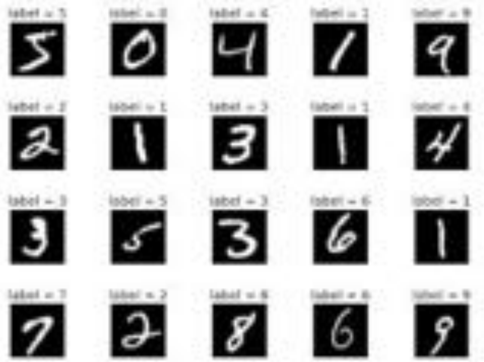
Due to time we will only watch 30 seconds of our reveal video describing the challenge which changes every year.

2018 Team 135 FRC Robot

Inspiration on the national level - The “cheesy poofs” #254

Machine Learning with Tensorflow and Keras

- Machine Learning defined and examples
- Moore’s Law
- How machine learning works
- Programming in Python
- What is Tensorflow?
- What is Keras?
- How does it all connect?
- What can we do with this?
- How can we share this?



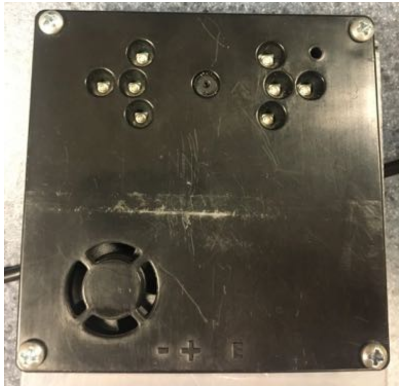
TensorFlow

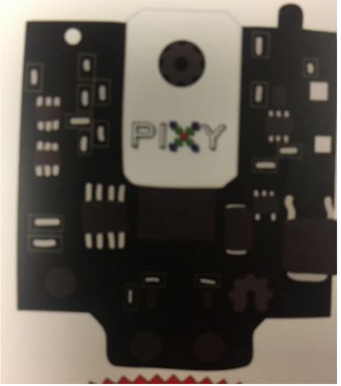
Keras

Computer Vision and Tracking in FRC

- Vision Tracking Hardware
  - Cameras and Coprocessors
  - Cell phones (FRC Legal Android Devices)
- Vision Tracking Software
  - GRIP
  - PixyMon
- Vision Tracking Programming
  - Implementation with Java through WPILibs and Eclipse
- Vision Tracking Applications and Limitations
- Vision Tracking Implementation
- Conner, Jim, and Kyle’s Summer progress:
  - [Code Available at Team135 GitHub]

Limelight vision





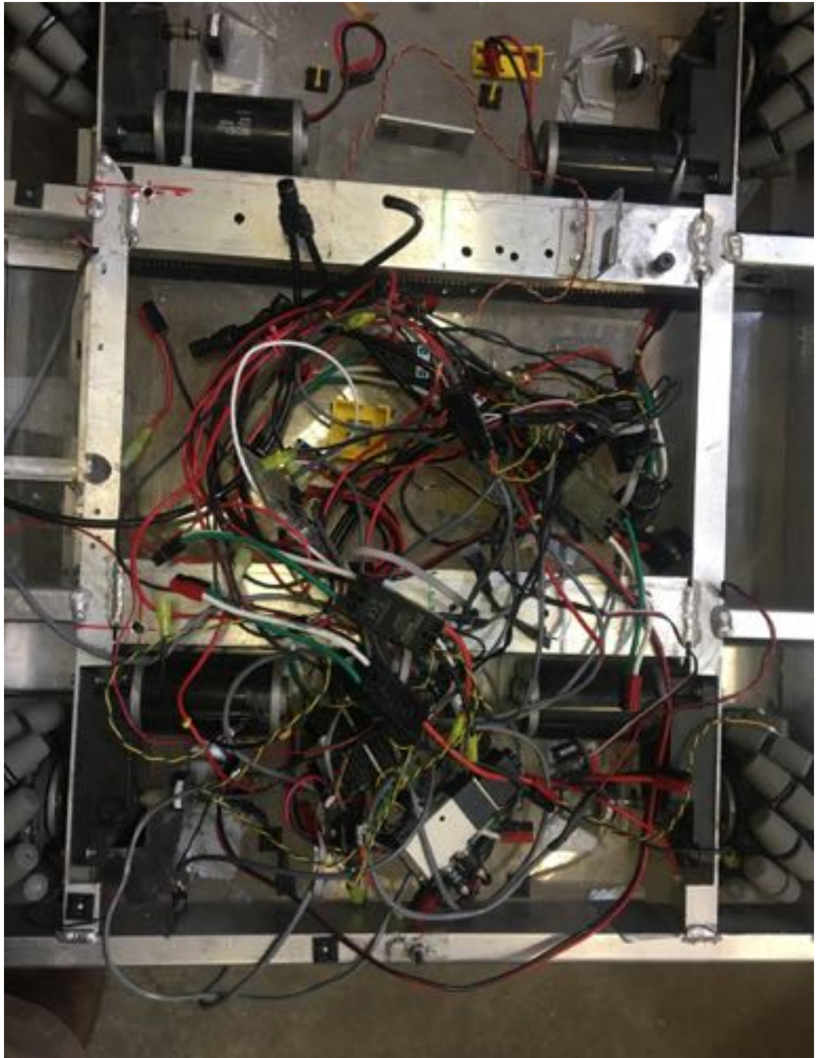
Pixy Cam

ZED Camera
With
NVIDIA Jetson TX2
processor

“There has be a better way”...

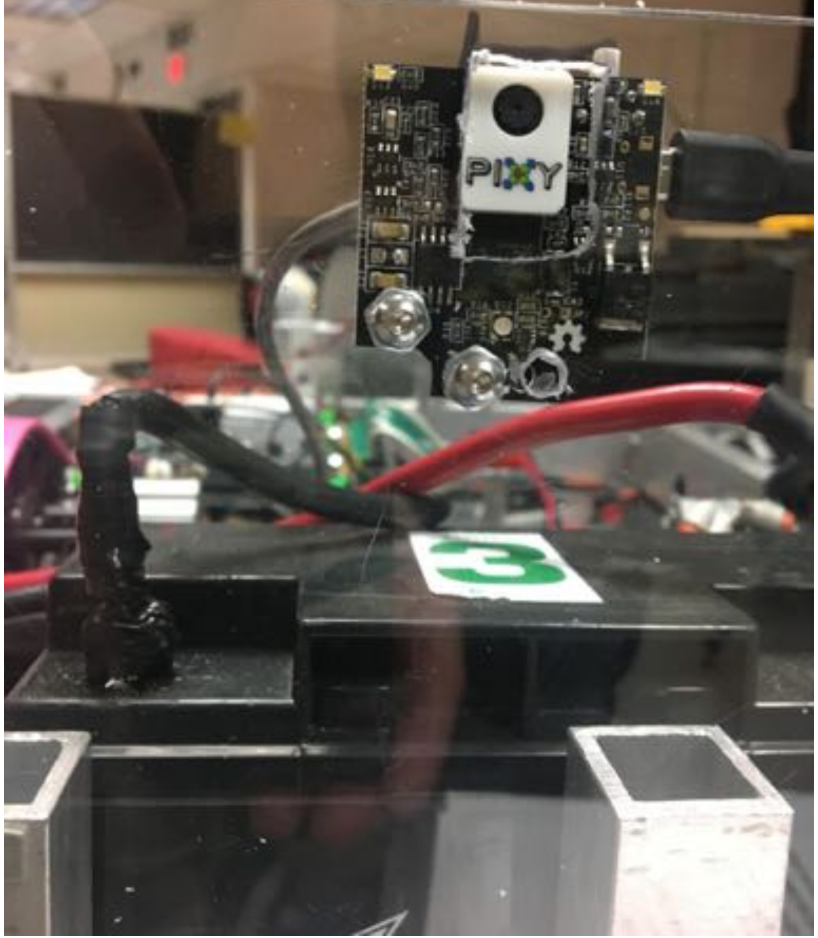
Kyle and I both knew we had to start at the beginning with this module. Teaching students proper names, practices, and uses with electronic hardware is a crucial step in the process. We are very grateful to be given the opportunity to develop this unit.

Areas of opportunity.
-Teach students proper wiring techniques.
-Teach students the impact of “modular design”.
-Create a system that can be easily moved to other robots for controls.
-The impact of efficient wiring.
-How efficiency can impact other sub teams.
-Ultimately we need a dedicated test bed for our software team to practice with.



Our project goal is to use one of 3 different camera systems to find a game pieces, move towards it, and hopefully increase our speed and accuracy.

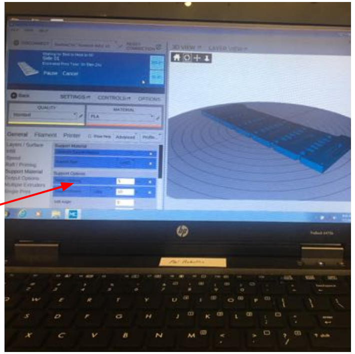
The Pixy Cam(right image) is the 1st camera system we are trying to implement. The yellow cube is the game piece we are trying to select.

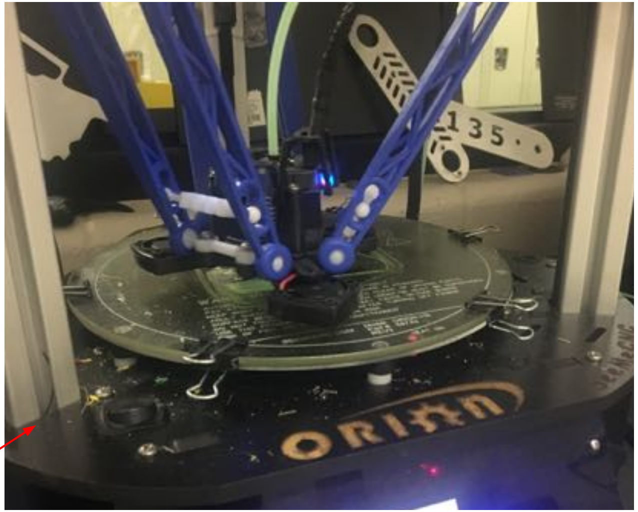


Thanks to FRC Team #27 -Team RUSH. we found files on thingiverse to print for our modular control board.

Hank hard at work!

3D print software

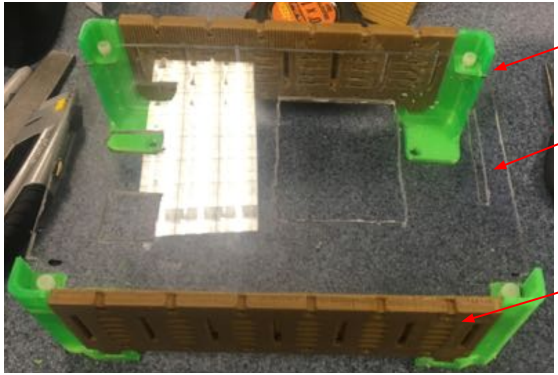




3D Printer

Sometimes 3D printing goes like this...

Finished Assembly

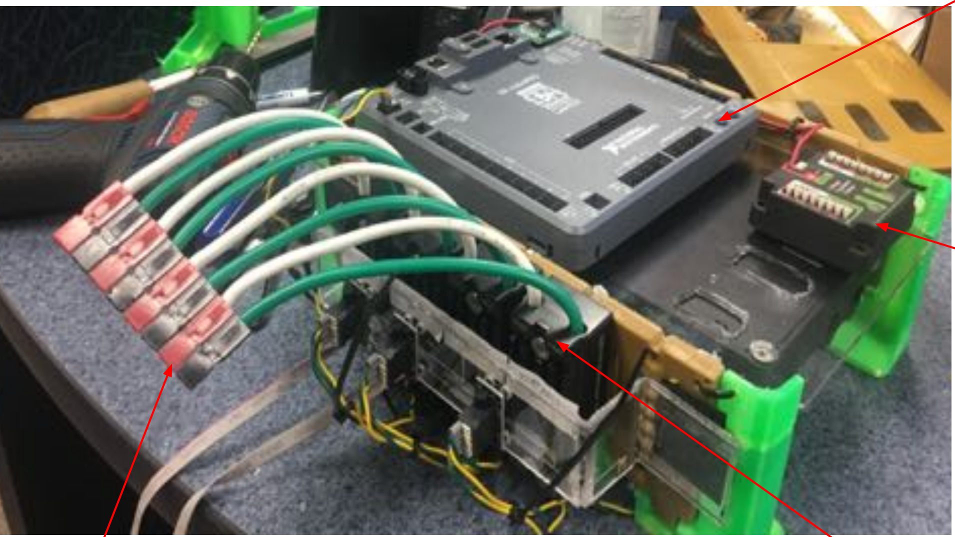


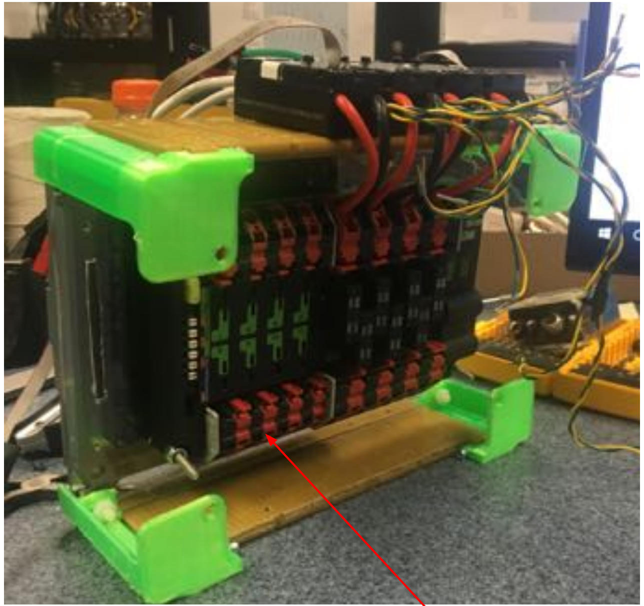
ABS towers

Lexann for mounting. Our printer couldn't handle the print size of this piece.

ABS side panels

Robo Rio -brain of robot





VRM-volt.
Reg.
module

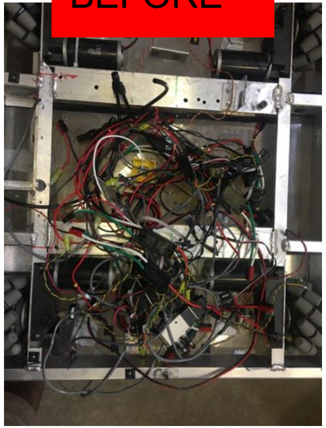
Connectors
going to motors

Speed
Controllers

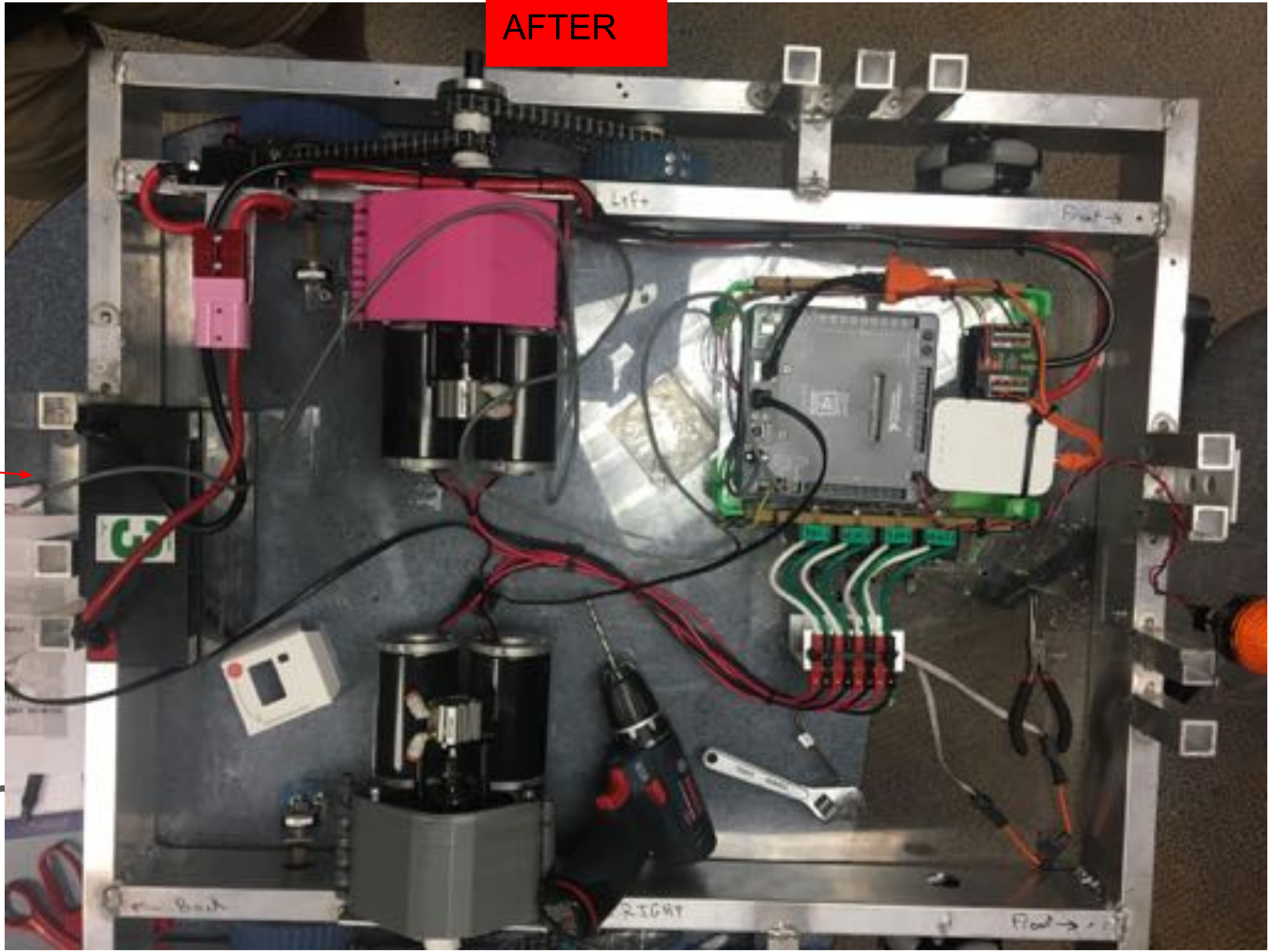
wiring the modular control board.

PDM-power distribution module. Supplying power from battery to entire robot on under side of module

BEFORE



AFTER

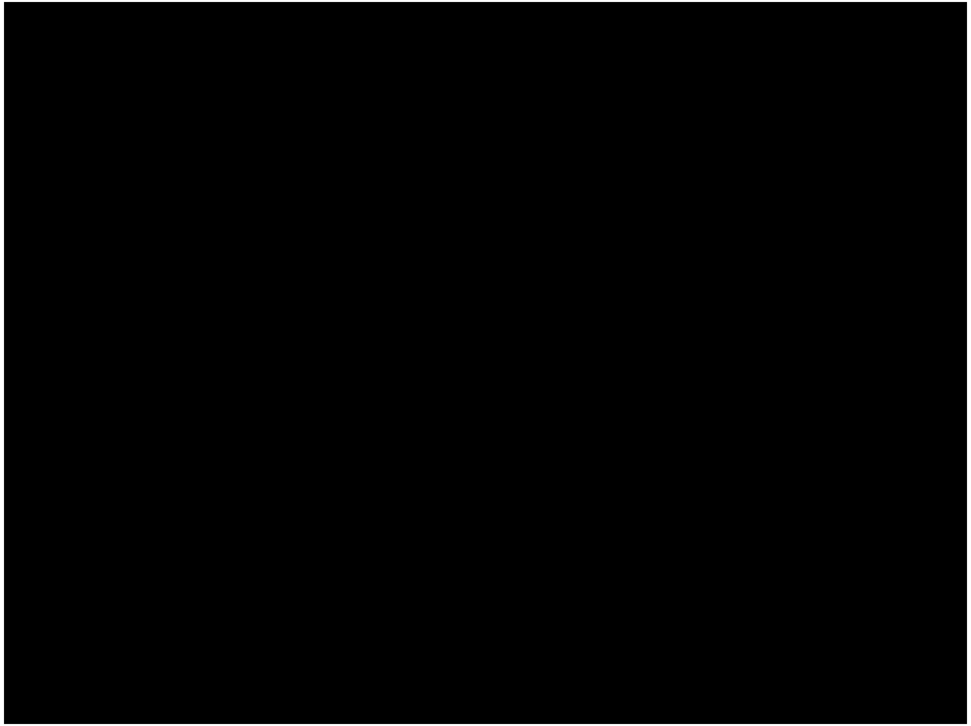


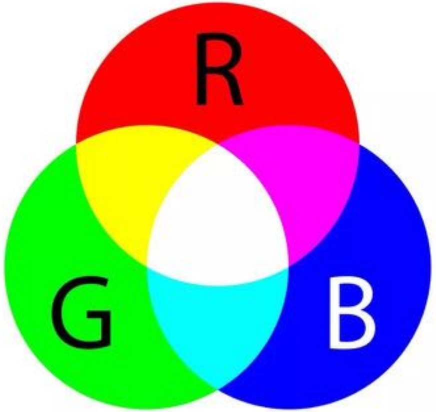
Robot battery

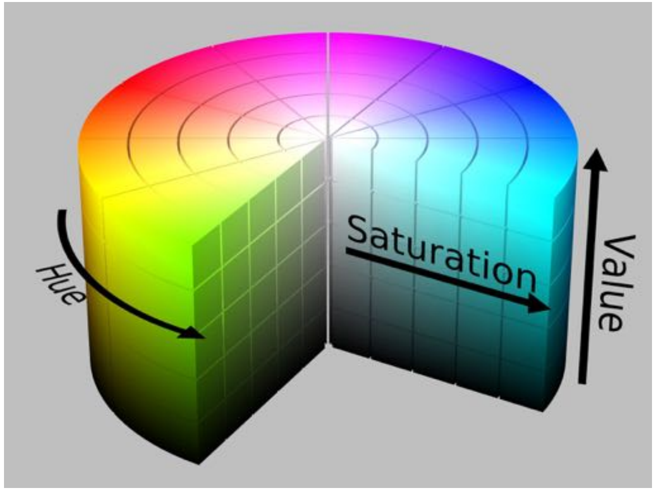
Pixy Cam mount

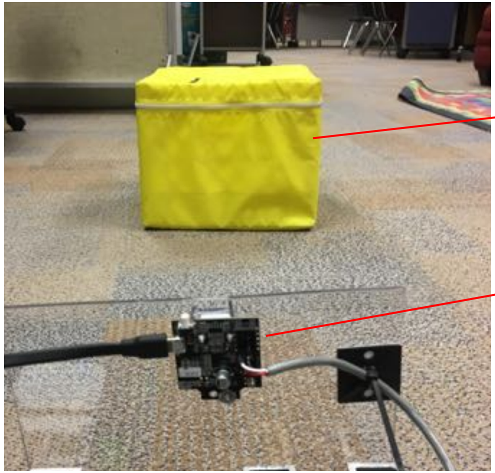
Finished test bed. Ready for software.

Tracking with GRIP and HSV



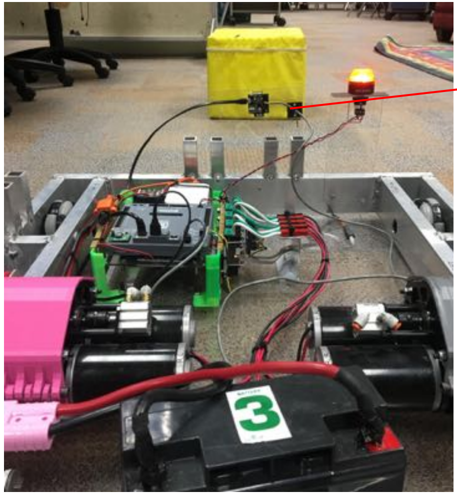




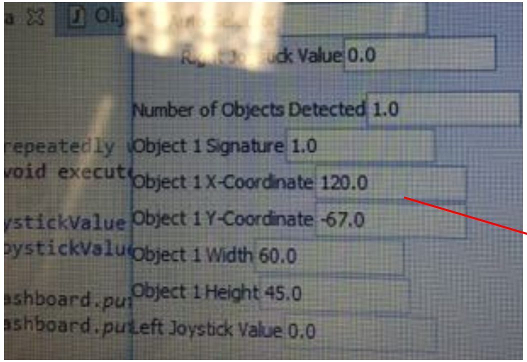


Target

pixycam



Pixycam mounted proudly on the front the robot.



The PixyCam is sending data to the roborio. Values are changing based on robot position to target! VERY EXCITING!!!

Value of X is changing in (+,- values) based on position of robot to left or right of target.

Our next challenge is controlling 4 motors to accurately steer robot to t target. We are very limite time in the game, so a constant challenge of spe and accuracy does exist.

Some of our 1st runs looked like this.

Our 1st successful run with camera locate and robot finding target.

This run shows the robot adjusting to the target being moved. It is working!

We are using PID (proportional integral derivative drive)for two functions

1. On the initial start up to find the cube in order to clean up the movement to find target.
2. On the movement to the target hoping to clean up the moves to correct left and right.

Module Activities

- Introduce Machine Learning as a topic, current applications, future applications, and Moore's Law
- Introduce students to machine learning hardware, software, and programming
- Students will learn about the various applications and limitations of vision tracking systems in FRC/robotics
- Students will work as a team to implement a vision system on a robot chassis that enables them to find a target, lock on to it, and autonomously drive the robot to the target, all without driver intervention.
    - Based on a Design Brief based on ML, AI, and vision tracking in large warehouses, such as those at Amazon.

Our Research

- ML in autonomous vehicles
  - NVidia is a major player in this field
  - Ford, Uber, Tesla, and GM all have dedicated ML teams
    - Outputs of autonomous vehicles are limited to three:
      - Acceleration rate
      - Deceleration rate (braking)
      - Turning Rate (yaw)
      - VERY similar to what we are hoping to accomplish: Driving based on an input - only we aim to crash into it, not avoid it!
- *The Amazing Ways Tesla is Using Artificial Intelligence and Big Data*

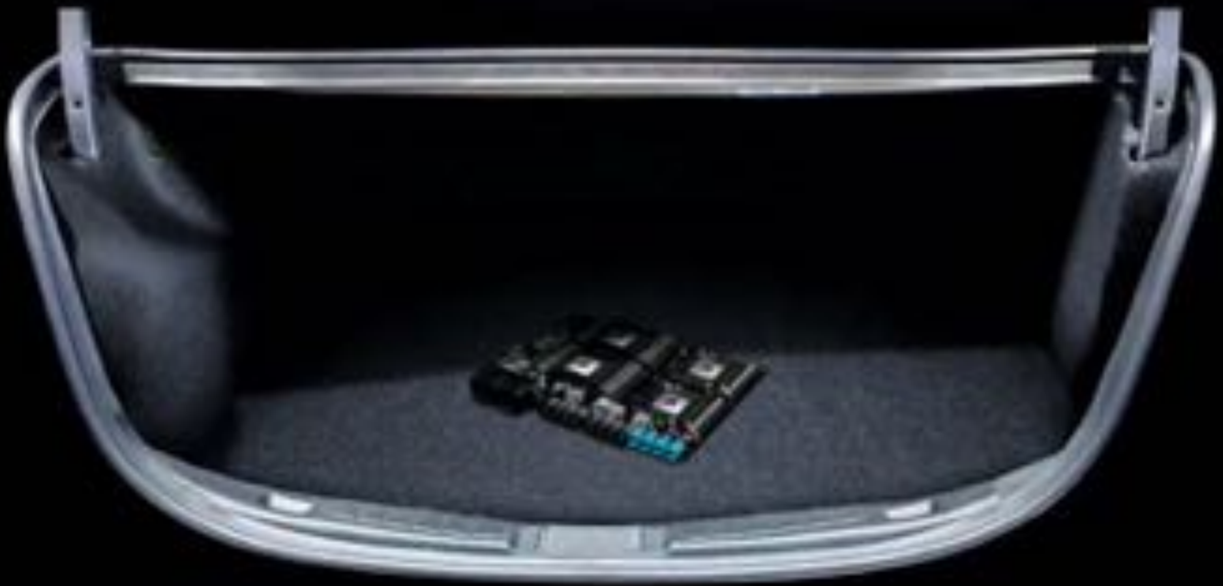
Nvidia Drive Pegasus

ANNOUNCING "PEGASUS"

ROBOTAXI DRIVE PX

320 TOPS CUDA TensorCore
16x GMSL | 4x 10G | 8x 1G | 16x 100M
Auto-grade | ASIL D
500W
Late Q1 Early Access Partners

Supercomputing Data Center in Your Trunk



More Research - Machine Learning in Robotics

- Computer Vision - robots “seeing” - This is where we based most of our work
- Imitation Learning - “reinforcement learning”
- Self-Supervised Learning - “enable robots to generate their own training examples in order to improve performance”
- Assistive and Medical Technologies
- Multi-Agent Learning

- https://www.techemergence.com/machine-learning-in-robotics/

Teaching with Tensorflow

- Tutorial on Teaching a basic robot with TensorFlow from Charlie Harrington at FloydHub
  - Possible inspiration for an additional module, given time
- Working on developing Cozmo as an introduction to robotics for Elementary students at Beiger, Machine Learning with TF for high school
  - “Adopt an engineer”

https://blog.floydhub.com/teaching-my-robot-with-tensorflow/

Our Keras Experiment

- Build our own dataset to classify images with red or blue spheres from the 2018 FTC Game, Relic Recovery
- Built on Keras using Tensorflow backend
- Had to resize all images, built a Python app to batch resize
- Created our own data set
  - 322 training images
  - 40 validation images
  - > 99% accuracy over 5 epochs
    - Need more and more diverse datasets for more realistic output
    - Needs some tweaking, but the core is there!

```
In [6]: runfile('C:/Users/marshk/ND_ML/ND_Keras_Experimental.py', wdir='C:/Users/marshk/ND_ML')
Tensorflow version: 1.9.0
Keras version: 2.2.0
Numpy version: 1.14.5
Image Width: 160, Image Height: 120
Found 322 images belonging to 2 classes.
C:/Users/marshk/ND_ML/ND_Keras_Experimental.py:87: UserWarning: Update your `Conv2D` call to the Keras 2 API:
`Conv2D(32, (3, 3), input_shape=(160, 120,..., padding="same")`
  model.add(Conv2D(32, (3, 3), border_mode='same',  input_shape=(160, 120, 3)))
Found 40 images belonging to 2 classes.
Epoch 1/5
20/20 [==============================] - 13s 674ms/step - loss: 0.4577 - acc: 0.8027 - val_loss: 0.2936 -
val_acc: 0.7812
Epoch 2/5
20/20 [==============================] - 13s 673ms/step - loss: 0.1754 - acc: 0.9375 - val_loss: 0.0280 -
val_acc: 1.0000
Epoch 3/5
20/20 [==============================] - 13s 626ms/step - loss: 0.0303 - acc: 0.9874 - val_loss: 0.0101 -
val_acc: 1.0000
Epoch 4/5
20/20 [==============================] - 13s 638ms/step - loss: 0.0498 - acc: 0.9875 - val_loss: 0.0236 -
val_acc: 1.0000
Epoch 5/5
20/20 [==============================] - 13s 632ms/step - loss: 0.0101 - acc: 0.9969 - val_loss: 3.5075e-04 -
val_acc: 1.0000
```

```
In [6]: runfile('C:/Users/marshk/ND_ML/ND_Keras_Experimental.py', wdir='C:/Users/marshk/ND_ML')
Tensorflow version: 1.9.0
Keras version: 2.2.0
Numpy version: 1.14.5
Image Width: 160, Image Height: 120
Found 322 images belonging to 2 classes.
C:/Users/marshk/ND_ML/ND_Keras_Experimental.py:87: UserWarning: Update your `Conv2D` call to the Keras 2 API:
`Conv2D(32, (3, 3), input_shape=(160, 120,..., padding="same")`
  model.add(Conv2D(32, (3, 3), border_mode='same',  input_shape=(160, 120, 3)))
Found 40 images belonging to 2 classes.
Epoch 1/5
20/20 [==============================] - 13s 674ms/step - loss: 0.4577 - acc: 0.8027 - val_loss: 0.2936 -
val_acc: 0.7812
Epoch 2/5
20/20 [==============================] - 13s 673ms/step - loss: 0.1754 - acc: 0.9375 - val_loss: 0.0280 -
val_acc: 1.0000
Epoch 3/5
20/20 [==============================] - 13s 626ms/step - loss: 0.0303 - acc: 0.9874 - val_loss: 0.0101 -
val_acc: 1.0000
Epoch 4/5
20/20 [==============================] - 13s 638ms/step - loss: 0.0498 - acc: 0.9875 - val_loss: 0.0236 -
val_acc: 1.0000
Epoch 5/5
20/20 [==============================] - 13s 632ms/step - loss: 0.0101 - acc: 0.9969 - val_loss: 3.5075e-04 -
val_acc: 1.0000
```

Impacts of RET Purchases

- PixyCam 2
  - We have developed software to use the PixyCam 2 to interpret information from video to drive/control robot output and behavior
- Jetson TX2
  - The Jetson is an AI development platform from NVIDIA that is specifically designed for deep learning applications. Students will use the Jetson to extend their abilities in Deep Learning and Machine Learning
- FRC Control Structure (MHS)
  - We purchased an FRC control system to use for development without cannibalizing one of our current robots so that we can further develop AI and ML concepts in robotics.

Questions?

Special thanks to Dr. Niemier and ND CSE for the opportunity to work together and develop this summer