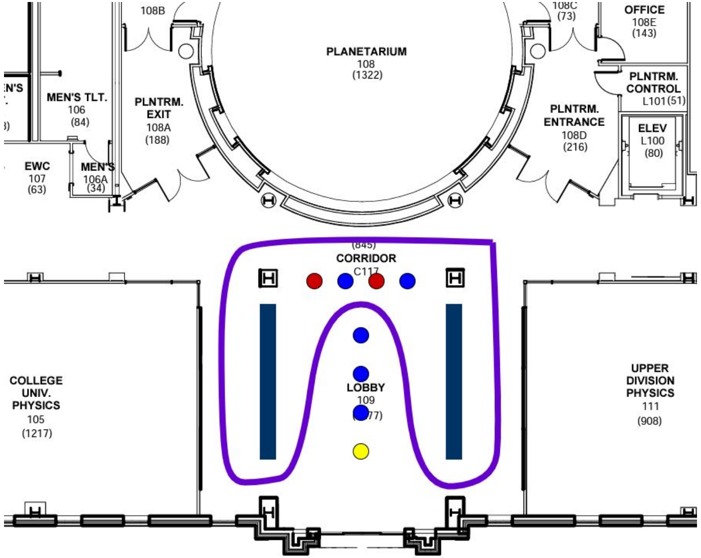
Senior Design Self Driving RC Car Final Presentation

Team FlashFire: Harrison Bounds, Brandon Donaldson,
Lawan Djibril Hamidou

Goal

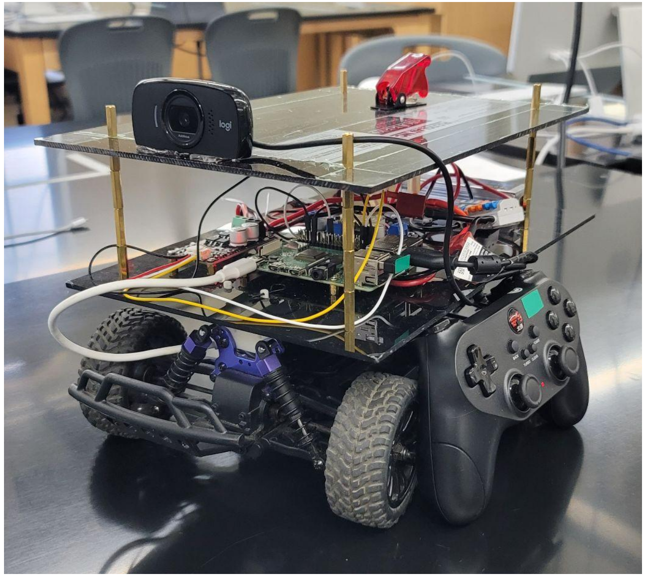
- **Navigate a course on campus using a trained autonomous model**



FIRST FLOOR
CONWAY CORP. CENTER FOR SCIENCES
SCALE: NONE

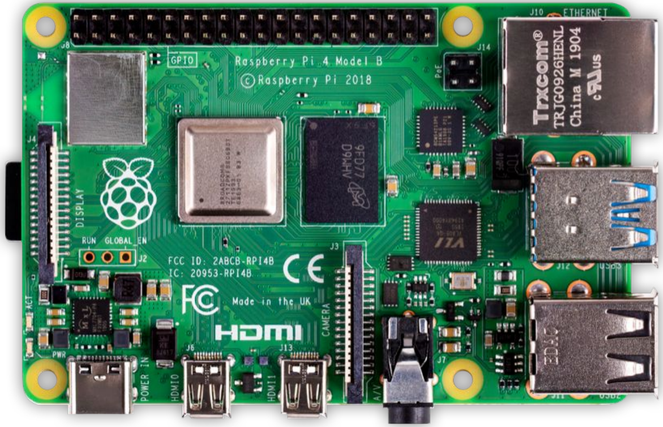
Steps

1. Build the Car
2. Collect Data
3. Train a Model
4. Deploy an Autopilot

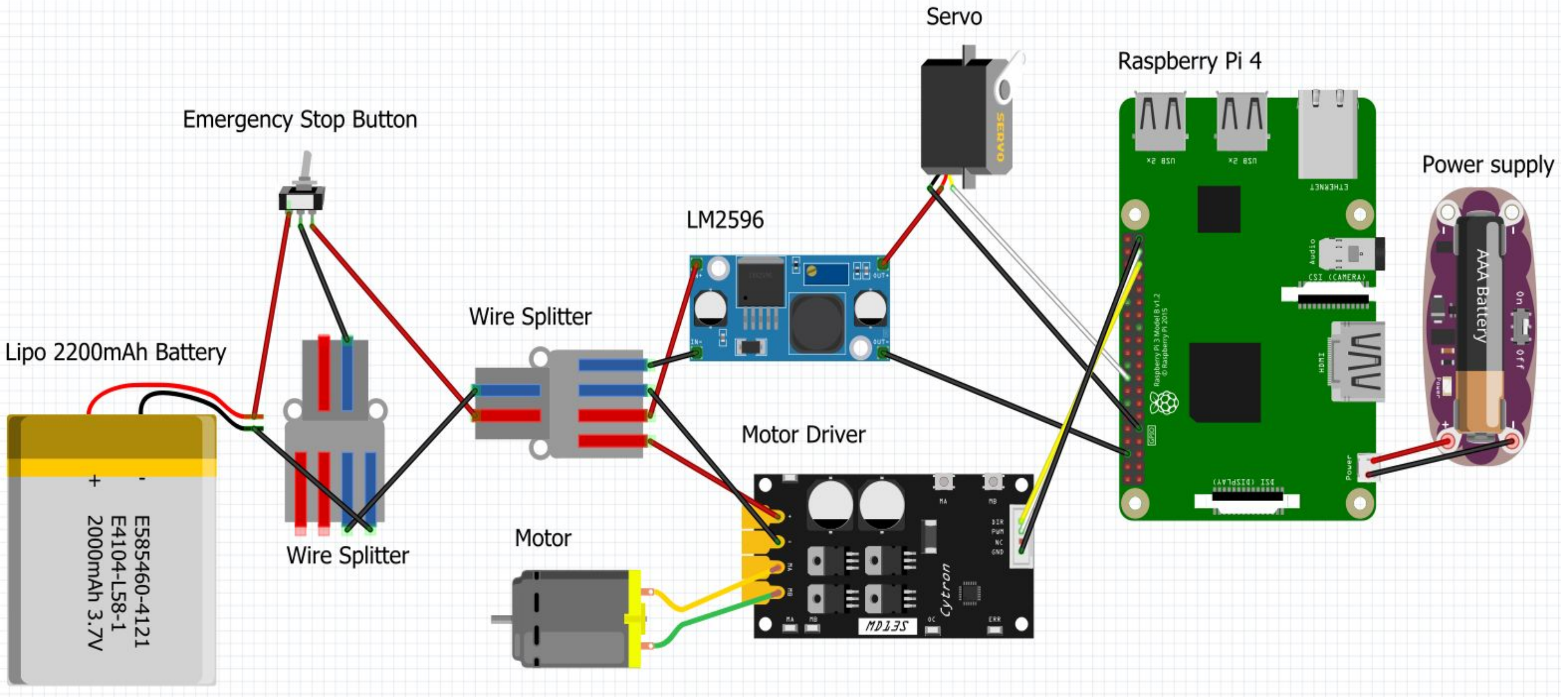


1 Hardware

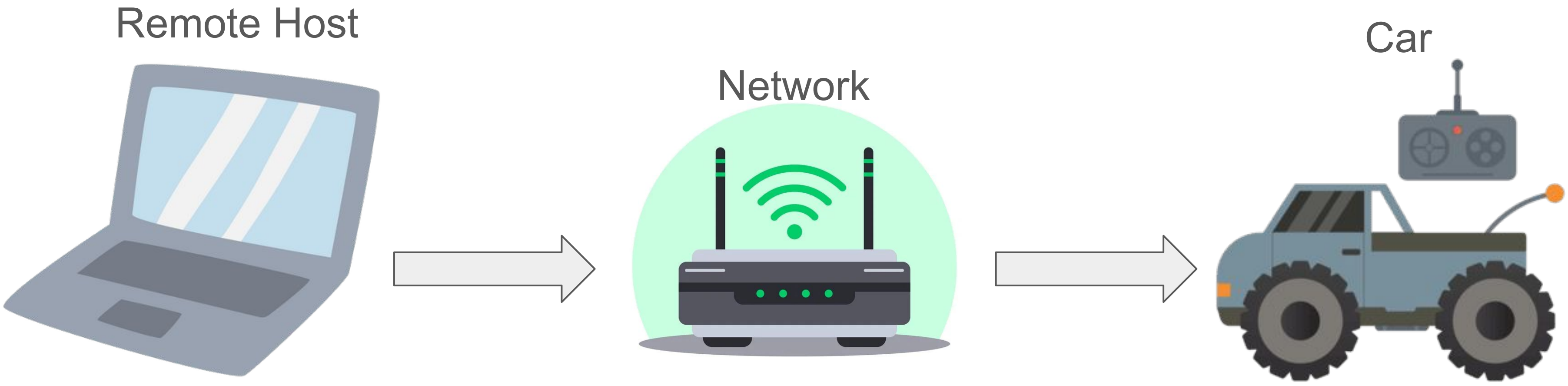
- USB webcam
- Raspberry Pi 4
- Voltage Regulator
- Motor Controller
- Powerbank
- LiPo Battery
- RC Car
- Shutoff Switch
- Controller



Wiring Diagram



Connecting to the Car



Remote Host

Network

Car

Driving the car

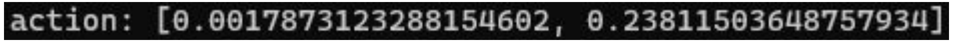
- Joysticks on controller send values to Pi
  - Full Forward Throttle = 1.0
  - Full Reverse Throttle = -1.0
  - Full Left Turn = -1.0
  - Full Right Turn = 1.0
- Joystick values can be scaled
  - newThrottle = realThrottle * 0.7
    - newThrottle will send a signal to the motor at 70% of the actual throttle input.

2

Collect Data

- Start the collect_data.py program
- Control the vehicle manually through the track
- Steering/Throttle Values are attributes for each image
  - Image is saved as a .jpeg
  - Values are stored in a .csv

Image that the car sees



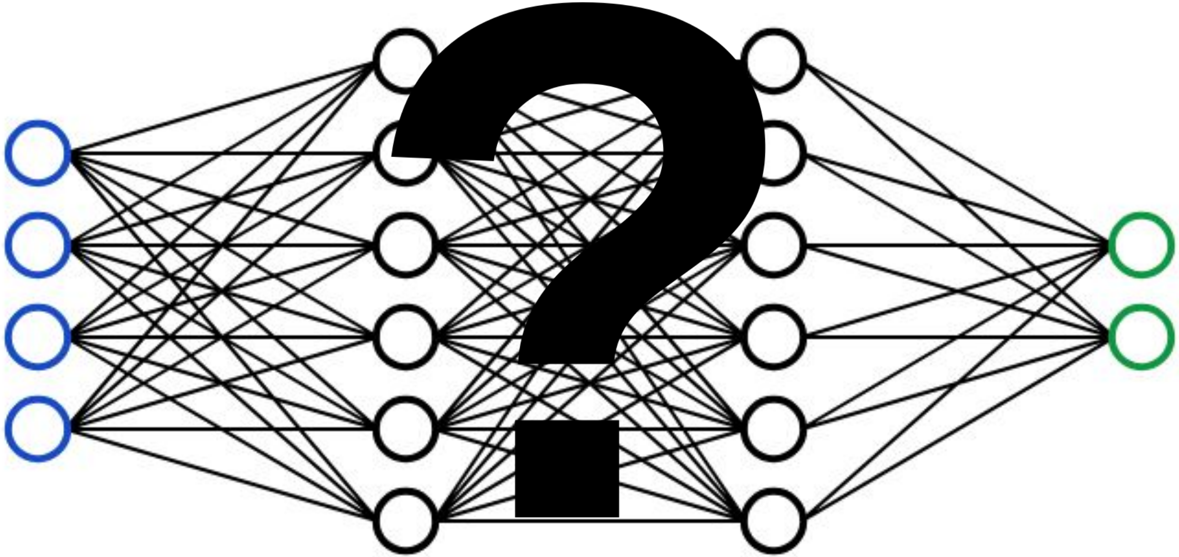
```
action: [0.0017873123288154602, 0.23811503648757934]
```

Steering, Throttle Value for image

Collect Data

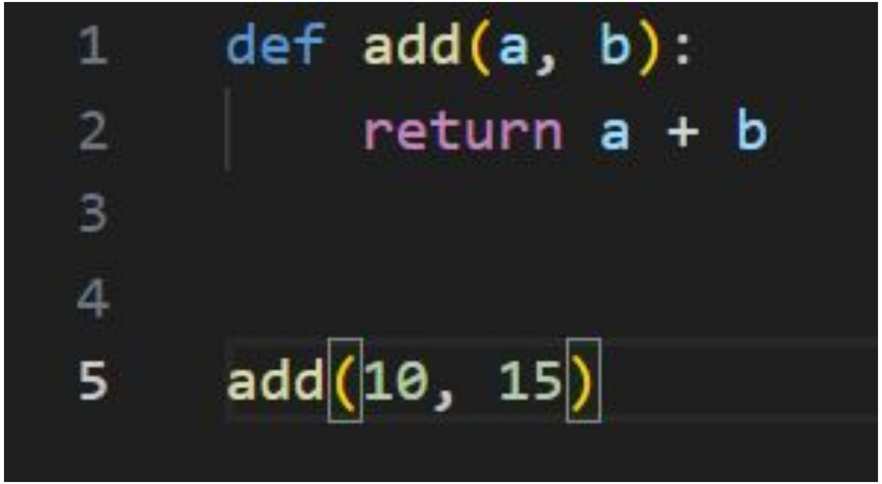
- Gathering ~20k-25k images
- Image size set to 120x160 pixels
- 15-25 laps (depending on size of track)
- Data is moved from RPi to be trained

3 Train a Model



What is a Neural Network?

$f(x)$

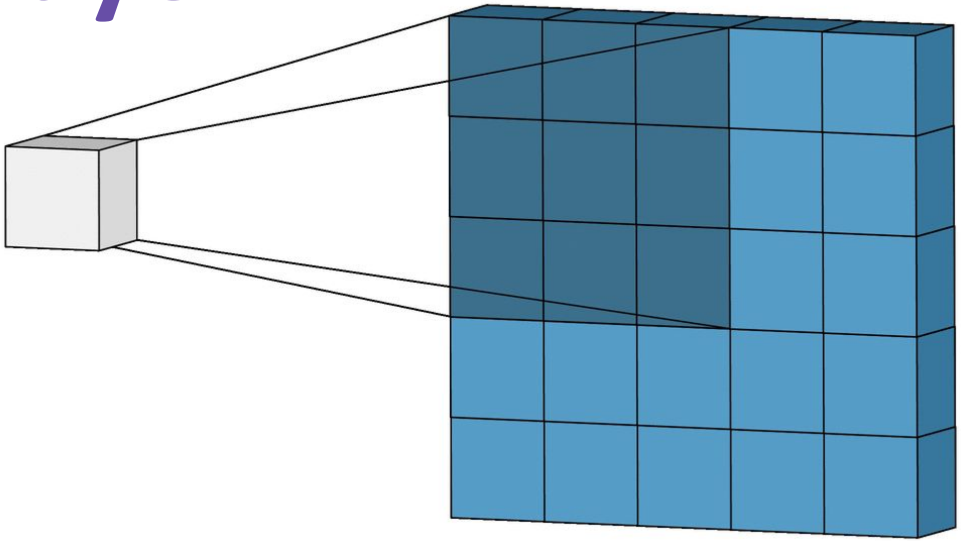


```python
def add(a, b):
    return a + b


add(10, 15)
```

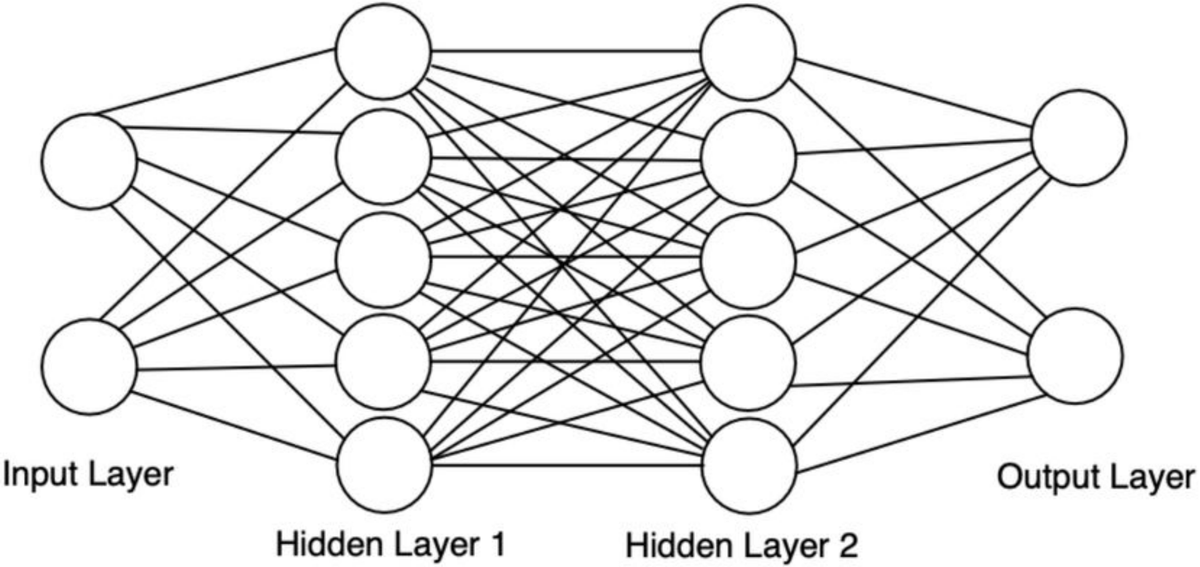
Convolutional Neural Network

1. Convolution Layer

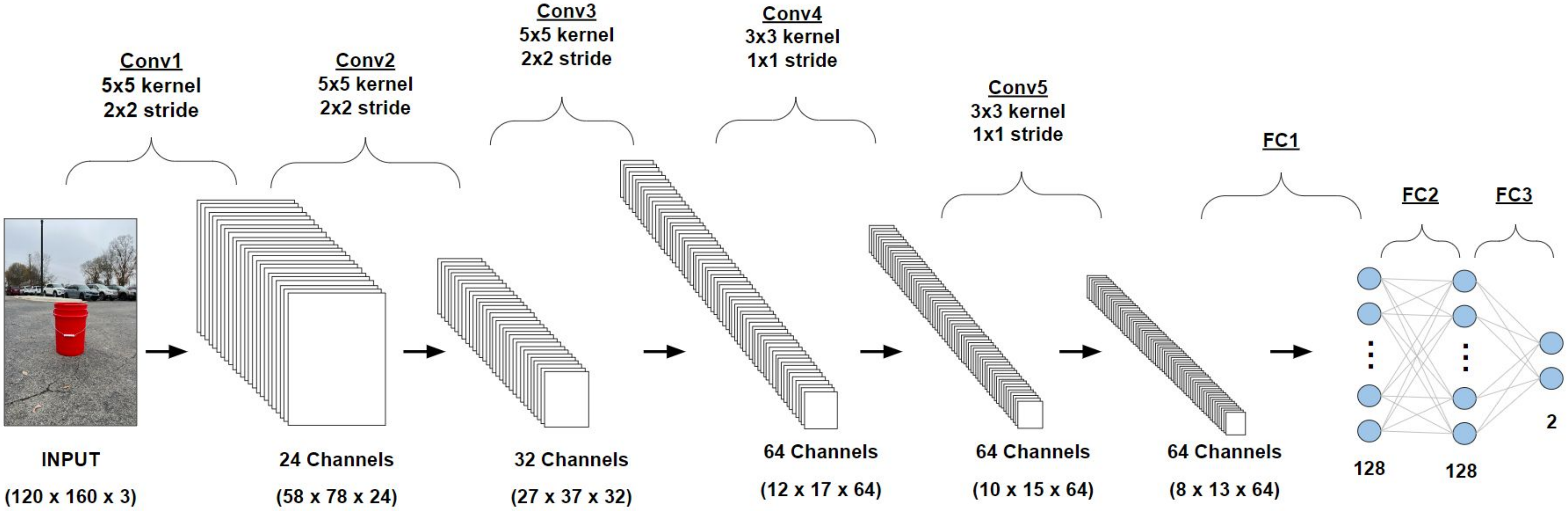


Convolutional Neural Network

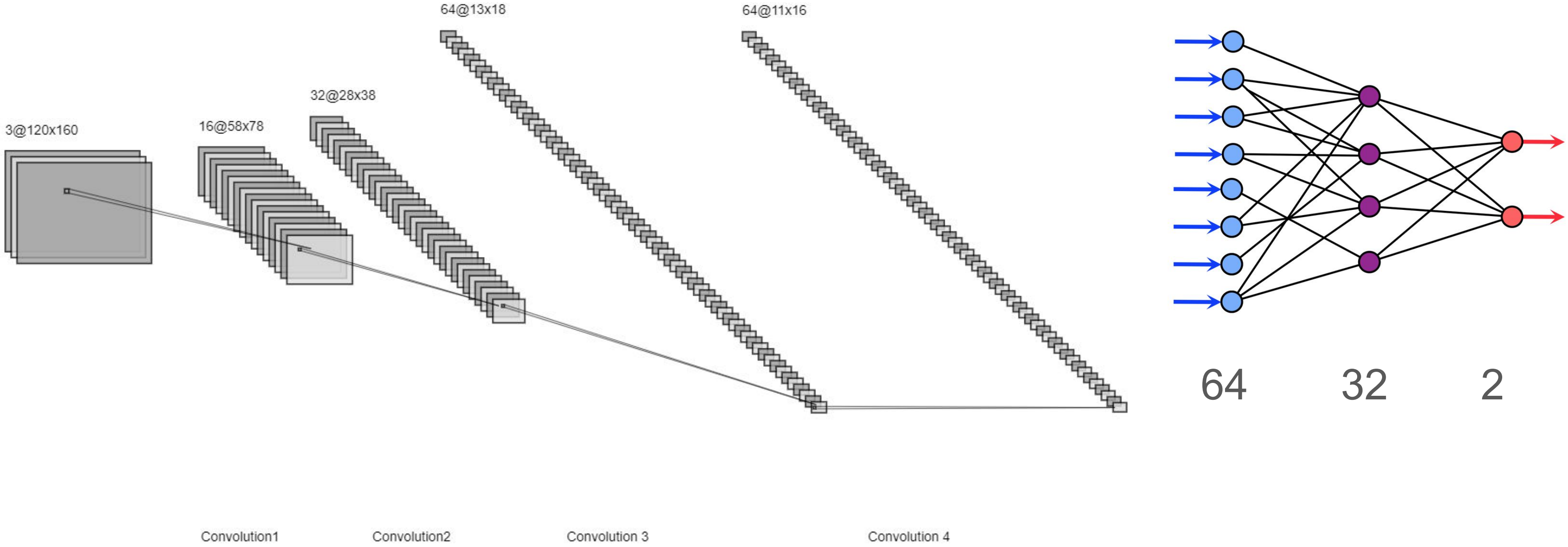
2. Fully Connected Layer

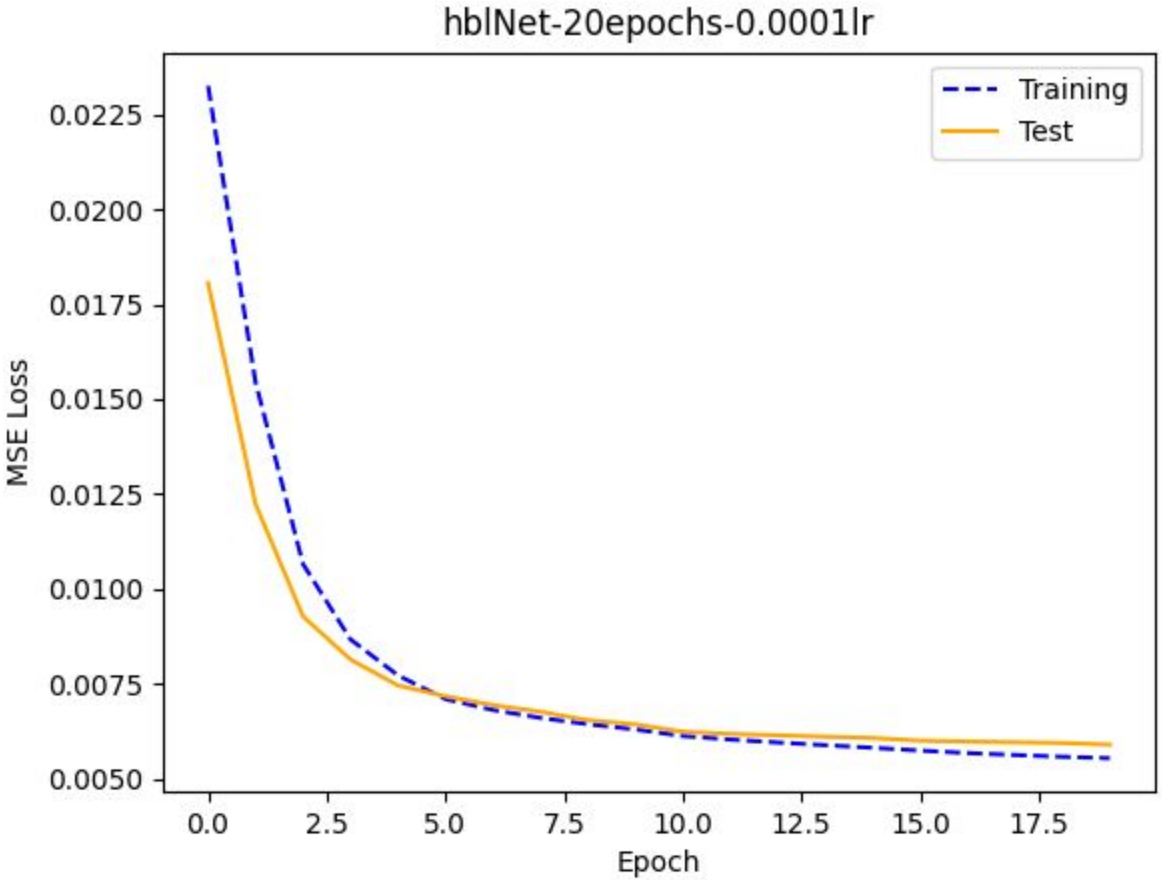


Original Model

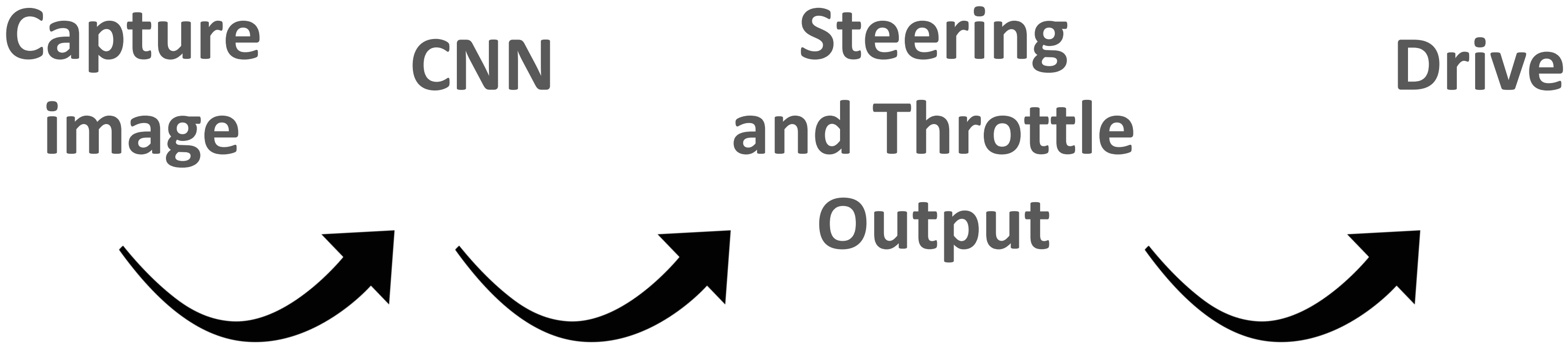


Our Model


64@13x18
64@11x16
32@28x38
3@120x160
16@58x78
64
32
2
Convolution1
Convolution2
Convolution 3
Convolution 4


hblNet-20epochs-0.0001lr
Training
Test
MSE Loss
0.0225
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch

4 Deploy an Autopilot



Capture image

CNN

Steering and Throttle Output

Drive

Conclusions

Results

- Created our own CNN architecture
- Developed a successful indoor autopilot
- Documented process and code for anyone to use

Issues

- Poor performance in the sunlight
- Time to collect data

Conclusions

Future Work

- Implement the Data Augmentation and Noise Injection for a more Robust training set (code already included)
- Test newer CNN architectures (ResNet, DenseNet)

THANK YOU