Smart Trap

Team 63
Jonathan Drugas, Xiaowei Yuan, Christian Morales
ECE 445

Introduction and Objectives

- Add-on for existing pressure plate activated traps
- Safe trap method to capture creatures like raccoons while leaving non targeted creatures like pets alone

How it Works

- Our device defeats the trap in idle state
- Servo physically blocks the pressure plate
- Camera & computer vision detect target animal
- Target detected -> servo moves -> trap armed

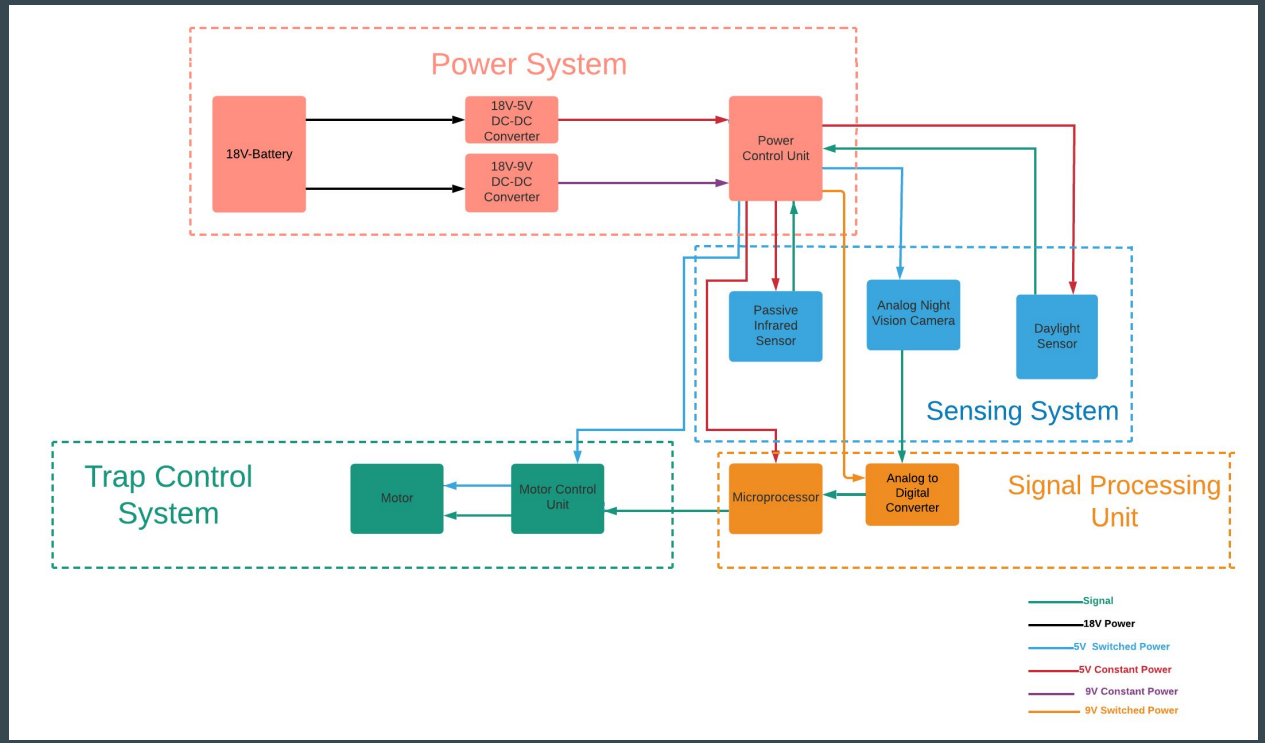
Ethics and Safety

- No animals were handled when testing this device
- Pressure plate traps are non-harmful devices

High Level Requirements for Smart Trap

- 12 hours of battery life
- Ten second decision time
- Pressure plate control
- System should turn off during the day

Block Diagram from Design Document

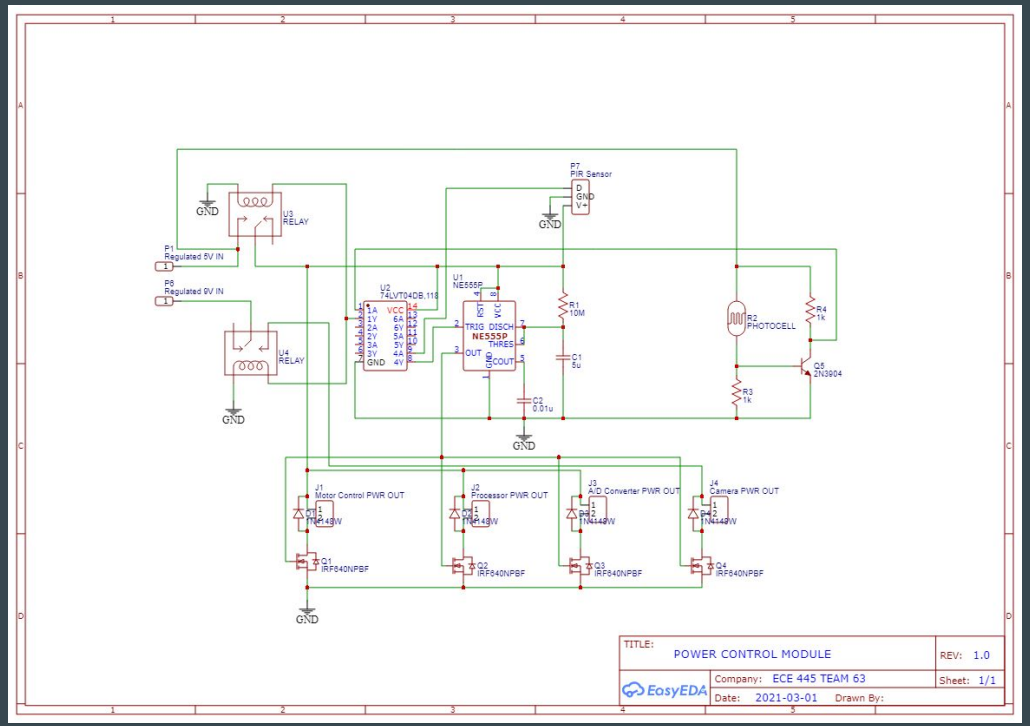


Design Process

Revisions to Initial Circuit Design (Power Control)

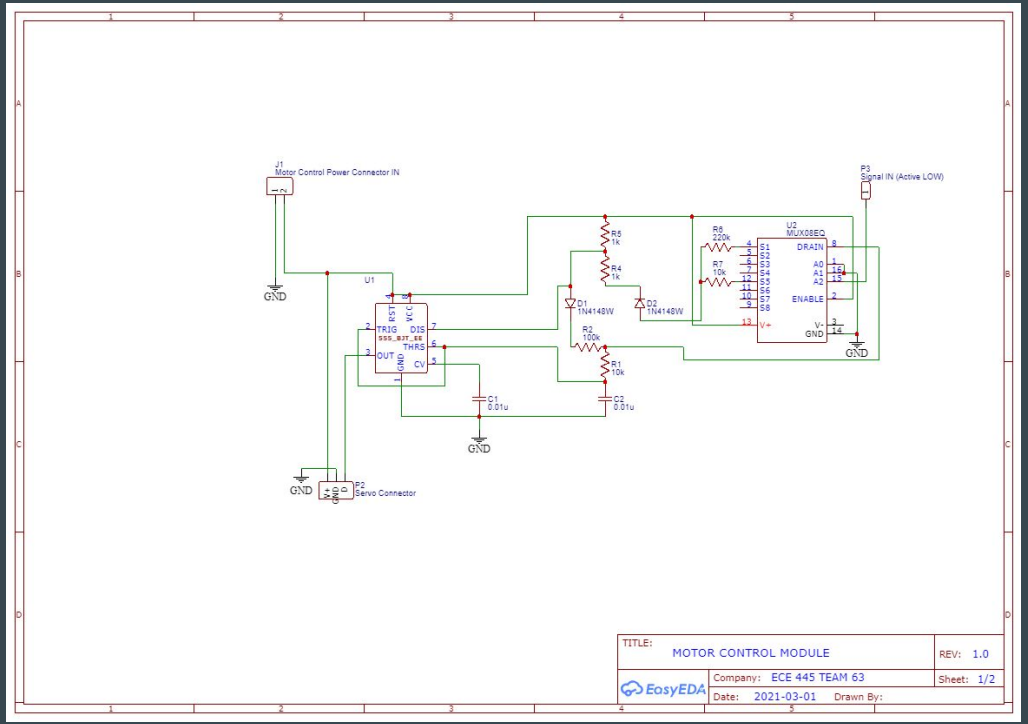
- Two main components:
  - Idle power switching
  - Daylight sensor power switching

- Revisions:
  - Q5 switched to NMOS
  - R3 switched to 10kOhm
  - Standard capacitor and resistor values were used (4.7uF instead of 5uF)



Revisions to Initial Circuit Design (Motor Control)

- This initial design did not work
- The design may have produced our simulation results if an analog mux was used
- The simulation results that we expected would not have controlled the servo
- Several debugging phases were necessary to redesign this circuit



Motor Control Circuit: Motivation

- Reserve computing power for image detection

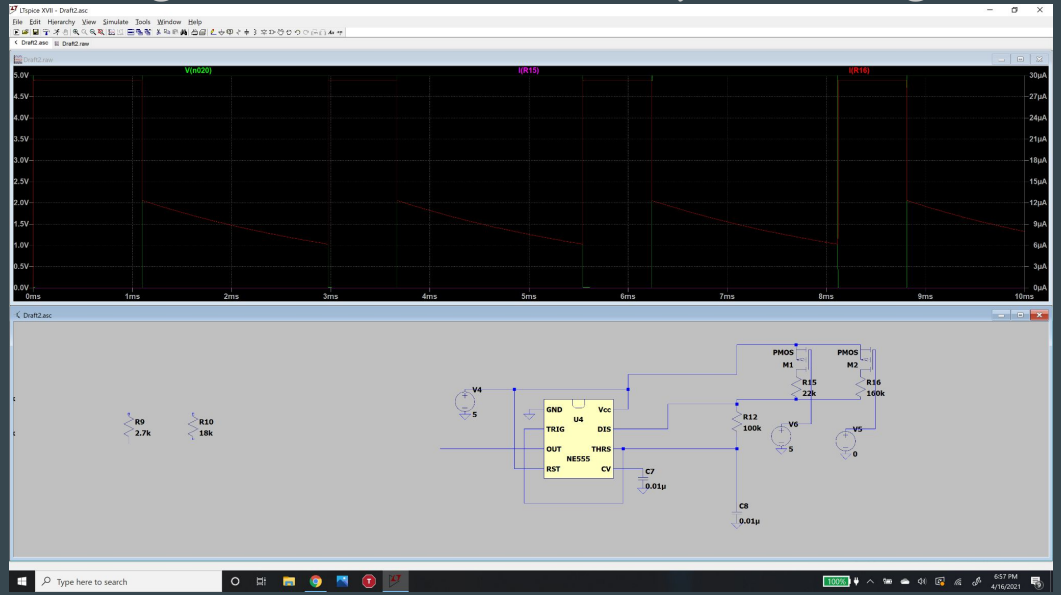
- ECE 445 guidelines suggest minimizing the use of microcontrollers

Motor Control Circuit: Debugging Process

- Debug using function generator
- Servo is responsive to pulse width in milliseconds, not necessarily duty cycle at a particular frequency
- 800us to 2.0ms controls the full range of motion

Motor Control Circuit: Debugging Process

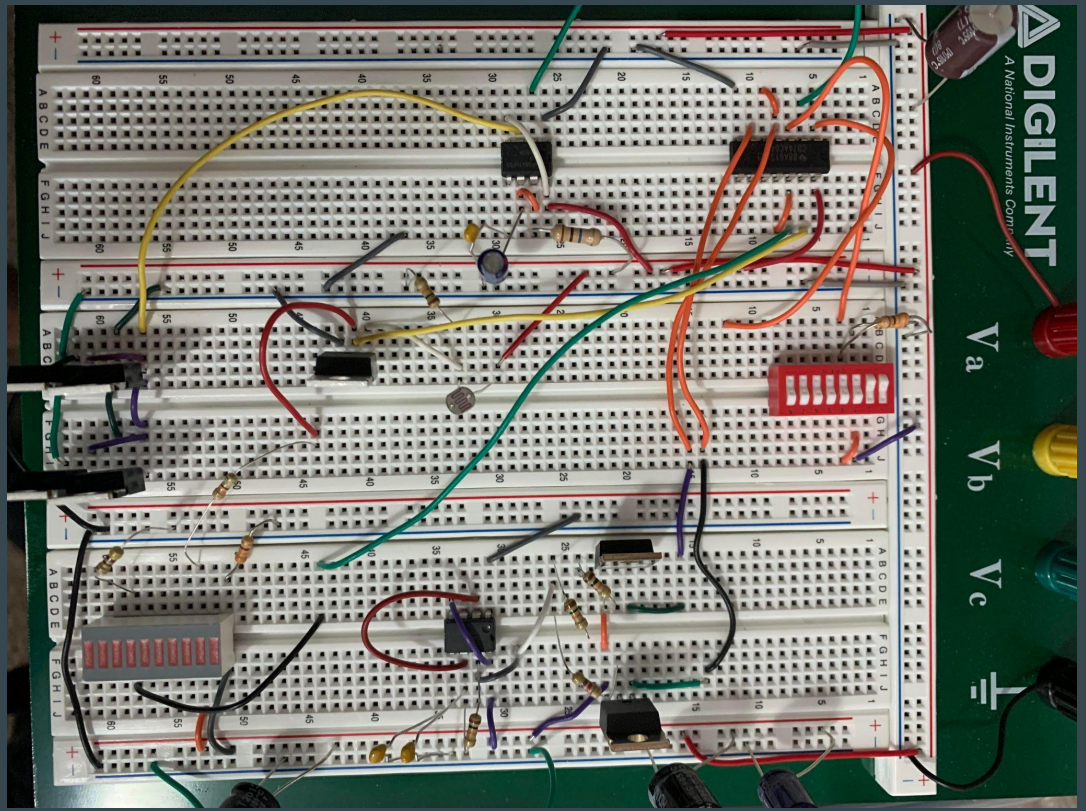
- LTSpice used for simulations
- How do we switch between pulse widths?
- Below is the working simulation. Currents only run through the switched PMOS.



Motor Control Circuit: Debugging Process

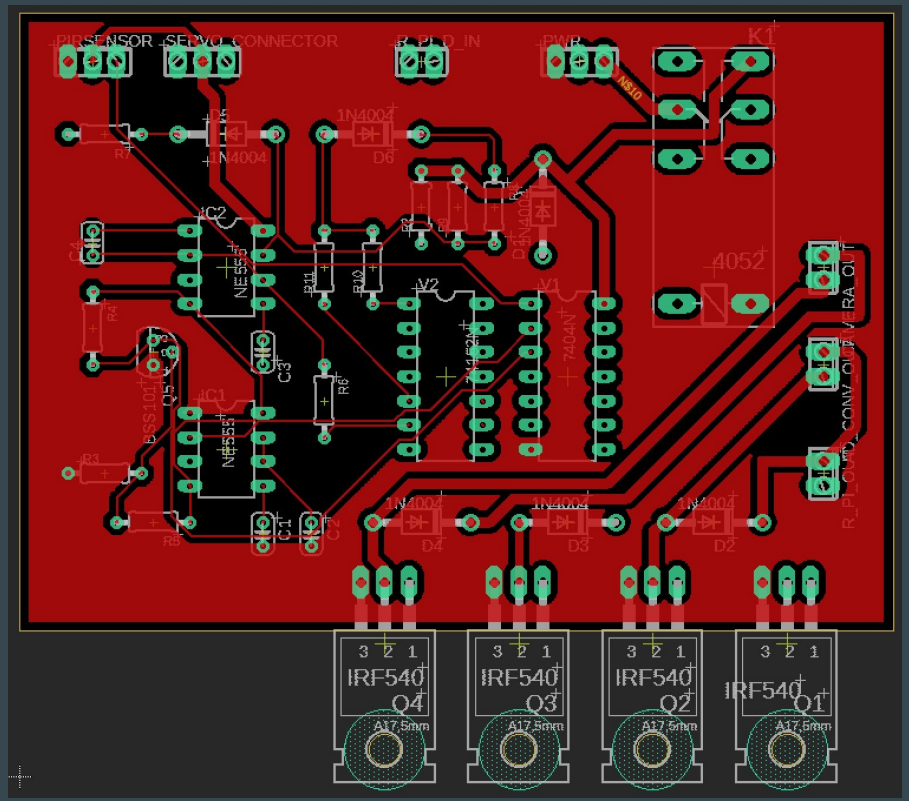
- Bench performance did not reflect simulation (Rds on from PMOS, etc.)
- Resistances were adjusted while looking at the scope
- Yellow represents motor control waveforms, blue represents pulse select

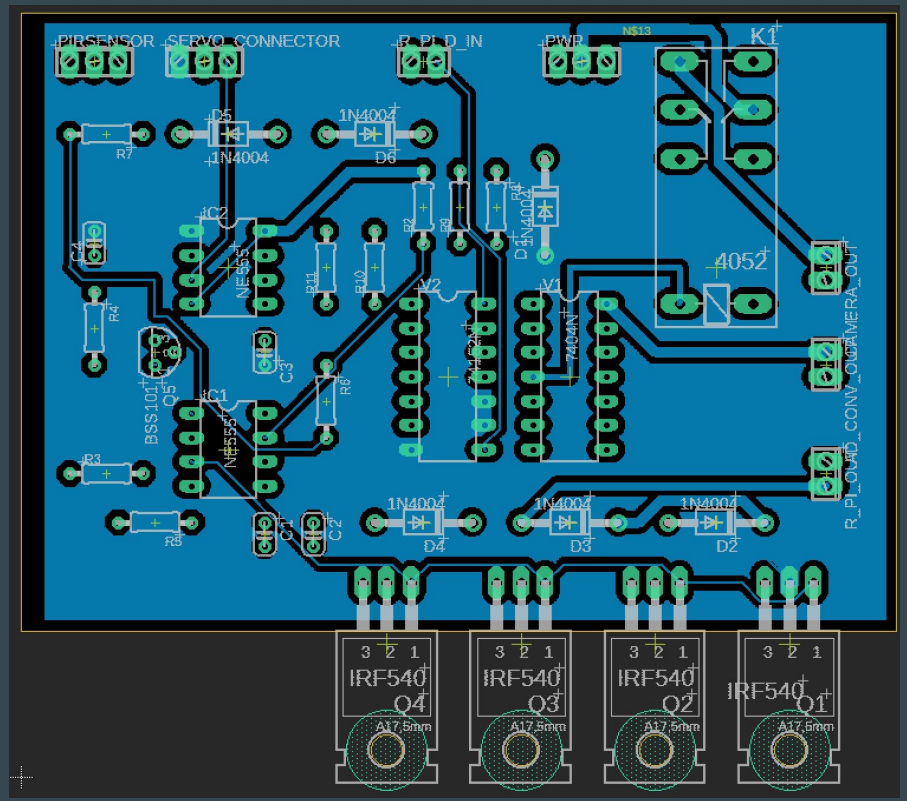
Current State of the Circuit on Breadboard



PCB Design Strategy

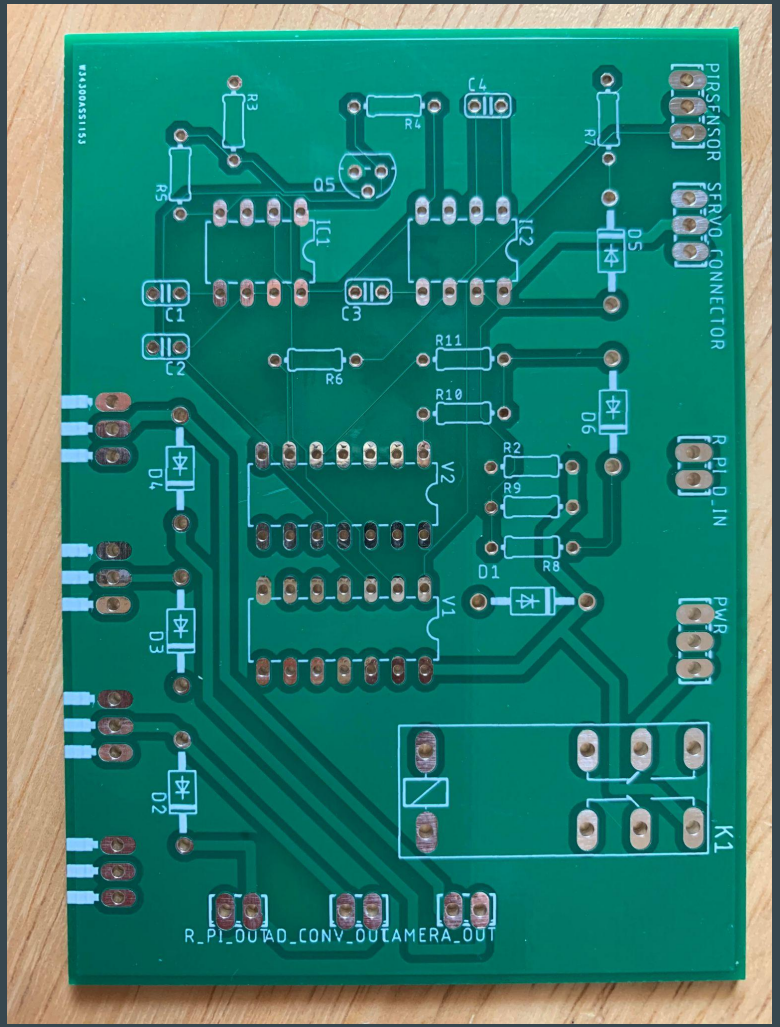
- The PCB was designed before the motor control was fixed
- Separate ground planes for sub-circuits


PIRSENSOR_SERVO_CONNECTOR
R_PI_D_IN
PWR
K1
4052
1N4004
NE555
7404N
BSS101
IRF540
A17.5mm
R_PI_QUAD_CONV_OUT
R_PI_CAMERA_OUT


PIRSENSOR_SERVO_CONNECTOR
R_PI_D_IN
PWR
K1
4052
1N4004
NE555
7404N
BSS101
IRF540
A17.5mm
R_PI_QUAD_CONV_OUT
R_PI_CAMERA_OUT

PCB Revisions

- Motor control topology should be updated
- Signal trace widths should be increased from 6mil to 10mil
- Some signal traces likely need better isolation

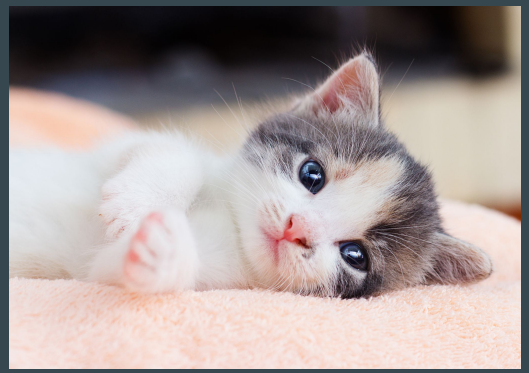


Computer Vision and Machine Learning

Machine learning : Goals

1. Animal detection

2. Animal classification

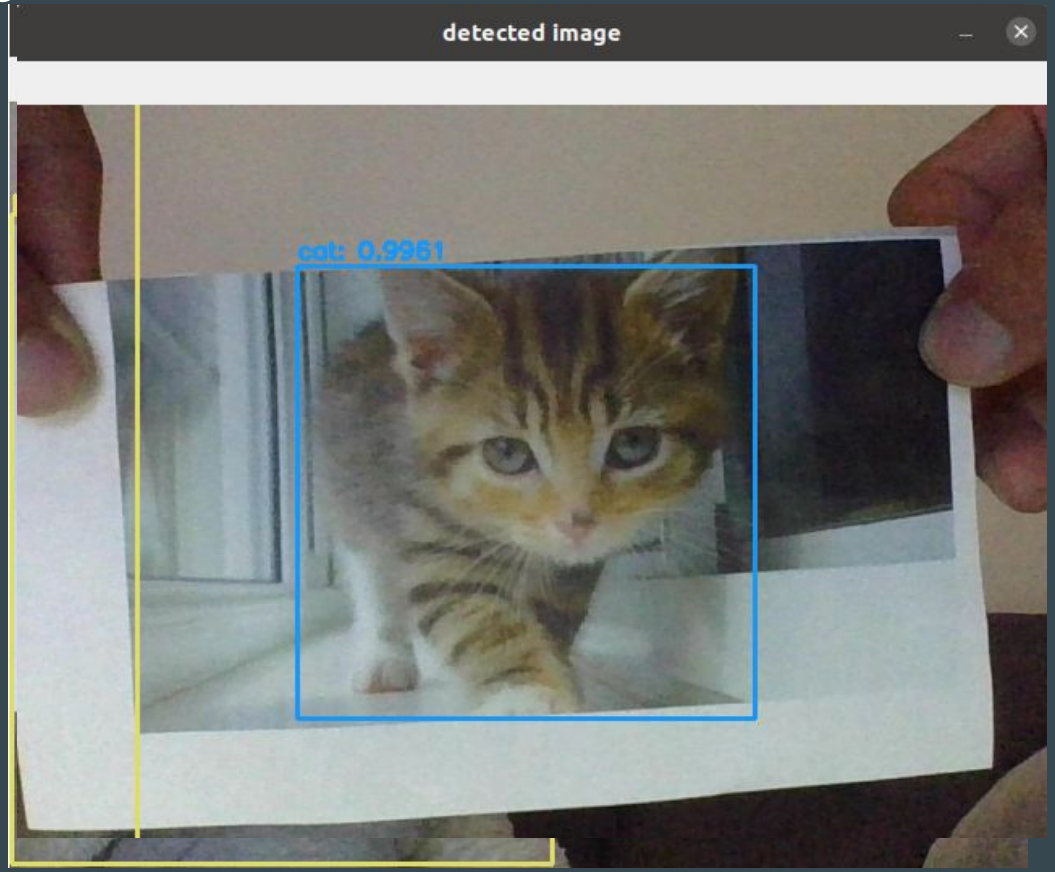


Machine Learning : Features

1. Fast

2. Accurate

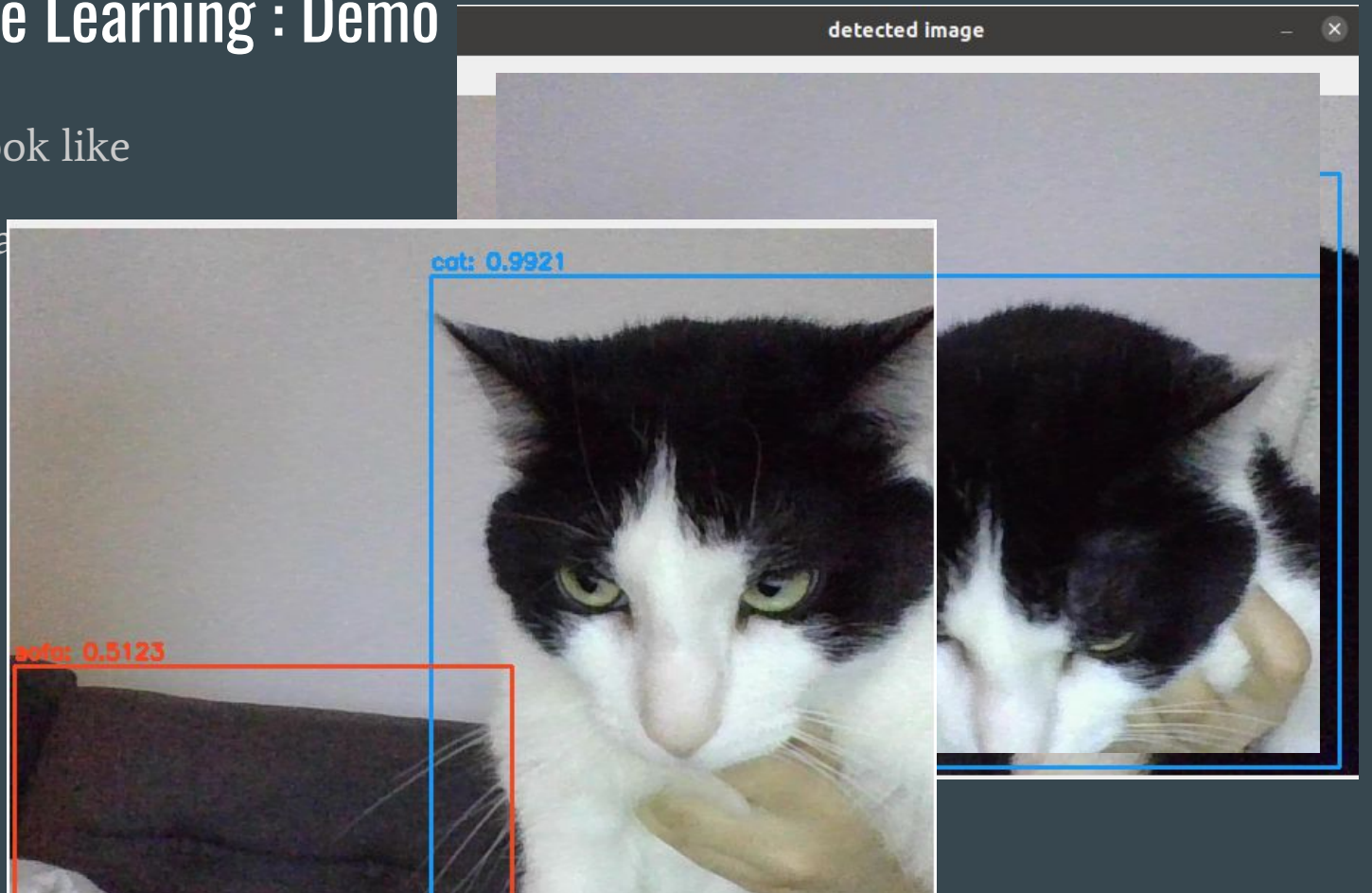
3. Run in real-time



Machine Learning : Demo

What's look like

if a real ca



Machine Learning : Running Environment

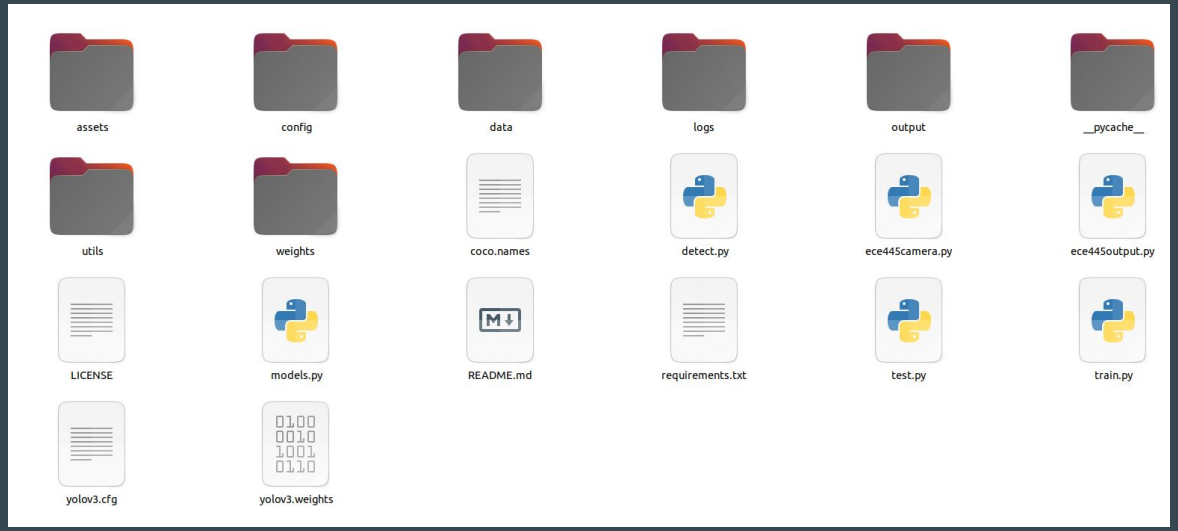
Linux (Ubuntu 16.04)

Python : 3.7.4

Tensorflow: 1.14.0

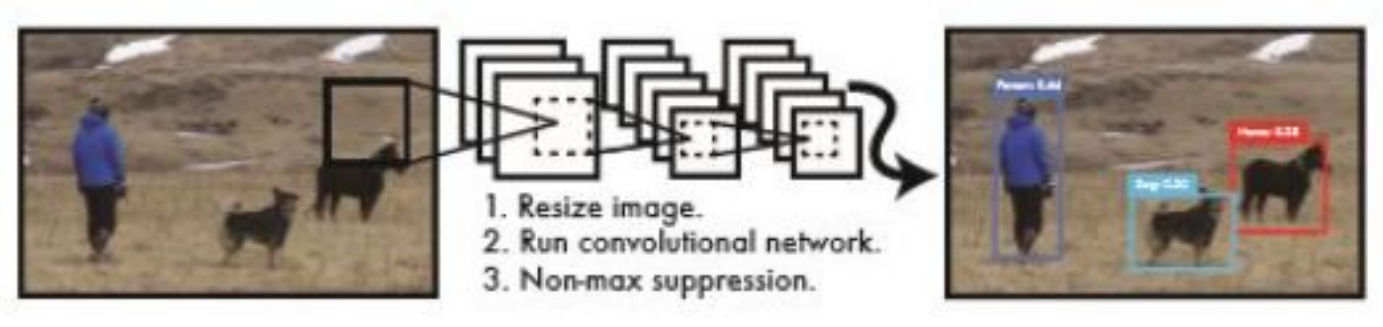
Keras: 2.2.4

Numpy: 1.17.4



Machine Learning : General Processing Steps

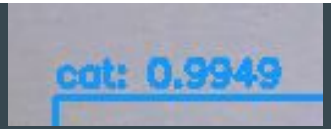
1. Resize
2. Convolutional layer
3. NMS



Machine Learning : Detailed Algorithms

Confidence value = $\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$

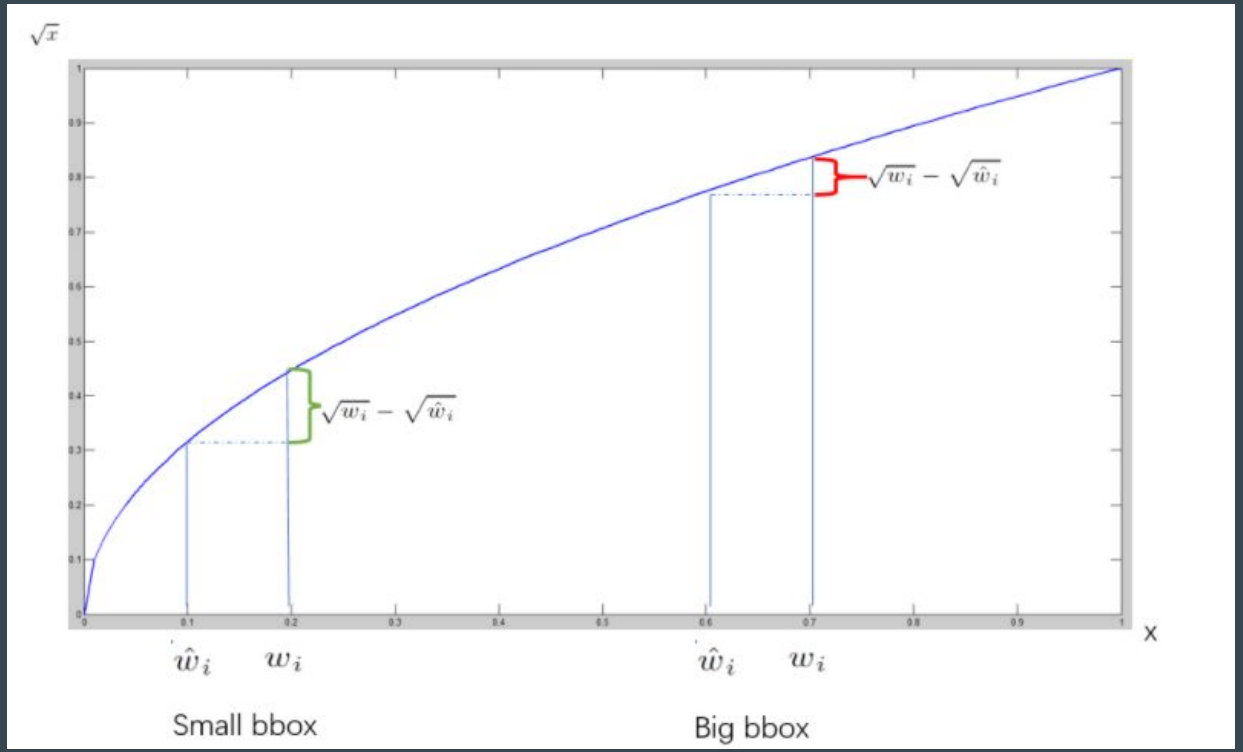
$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$



Machine Learning : Error Calculation

X: size of bounding box

Y: accuracy of prediction



Machine Learning : Real-Time Camera & Timing

```
cameraCapture = cv2.VideoCaptu                                    h camera we need to use
cv2.namedWindow('detected imag
cv2.setMouseCallback('detected
print ('for ECE445 ML DEMO')
success, frame = cameraCapture
while success and cv2.waitKey(
    # load video into Binary L
    blobImg = cv2.dnn.blobFrom

    net.setInput(blobImg)  # u


    outInfo = net.getUnconnect                                    t layers
    start = time.time()
    layerOutputs = net.forward
    end = time.time()
    print("[INFO] Our SmatTrap                                    # print info
```

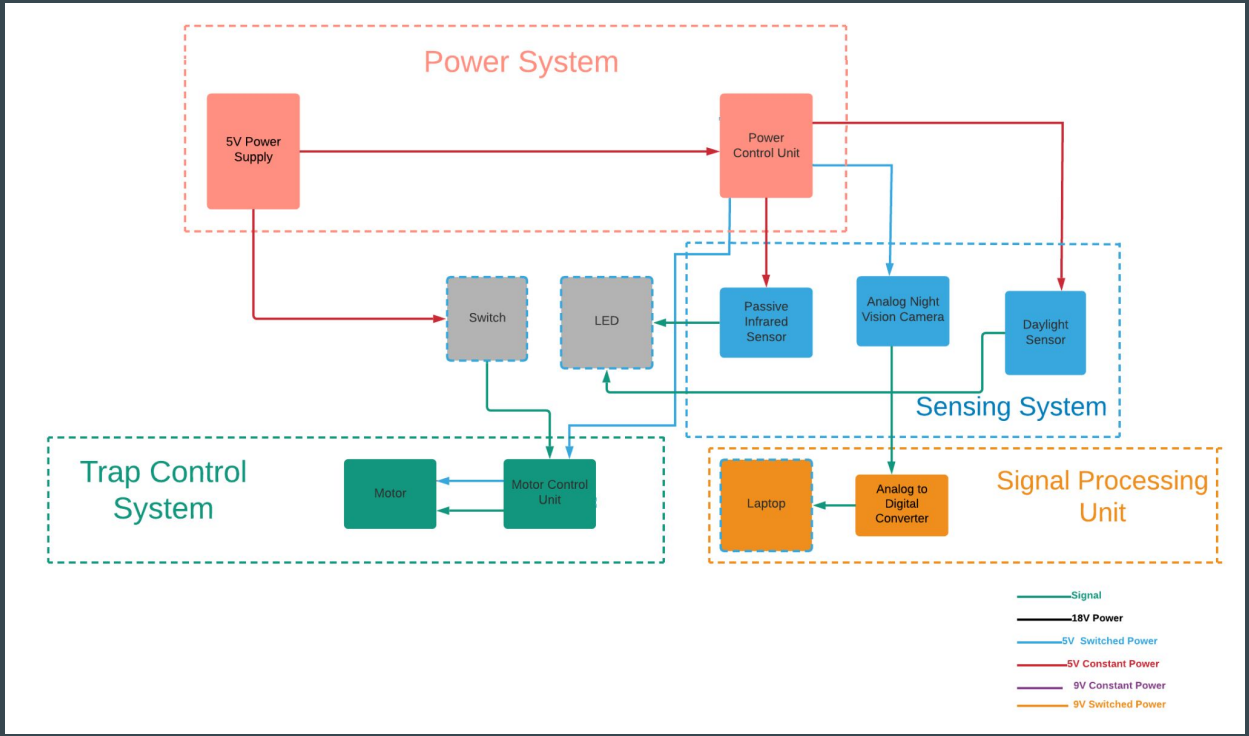
```
[INFO] Our SmatTrap ML took 0.167678 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.165931 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.164086 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167707 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.168516 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.168392 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167460 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167712 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.169045 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.170969 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.180642 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.175180 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.180270 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.176859 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167384 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.166099 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.163617 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.162779 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.167140 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.161730 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.161266 seconds to recognize this frame.
[INFO] Our SmatTrap ML took 0.169407 seconds to recognize this frame.
```

Machine Learning : More Improvement

1. Neural network & locally
2. Daytime & nighttime

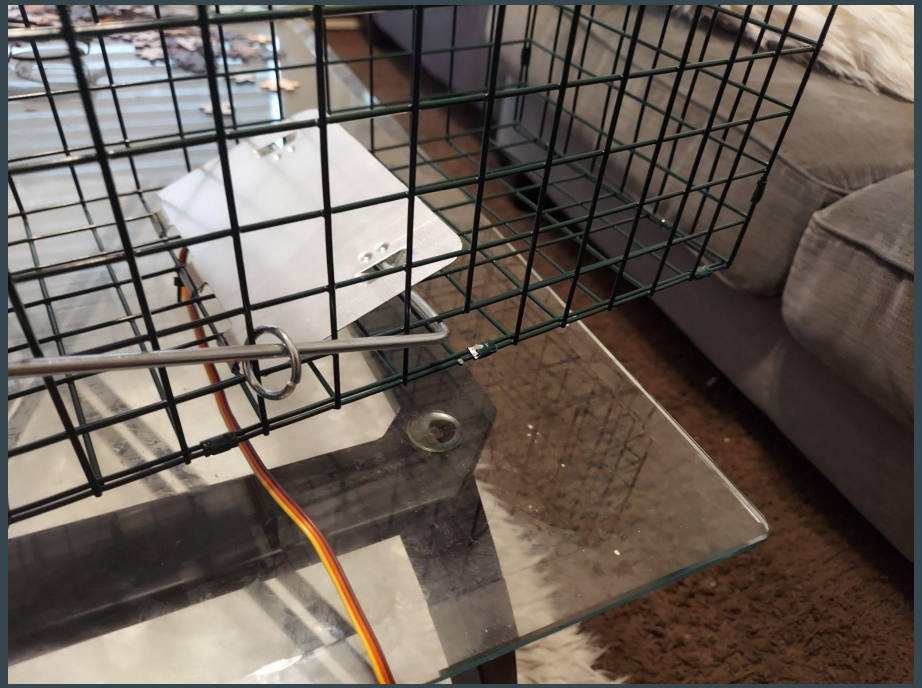
Current State of Product & Demonstration Results

Block Diagram from Demo (Debugging Stage)



Smart Trap Images

Smart Trap Images



Smart Trap Images

Smart Trap Images

Smart Trap Images

Smart Trap Images

Demonstration Results

- Proximity detection and component switching
- Pressure plate control
- Computer Vision

Further Work

- Battery power
- Daytime power savings
- Microprocessor
- Integration
- PCB

Thank You!

Any Questions?