FAMU-FSU College of Engineering

Pole Health Detection Sensor

Florida Power and Light

Team 301

Corie Cates

Alonzo Russell

Leonardo Velazquez

Thomas Williams

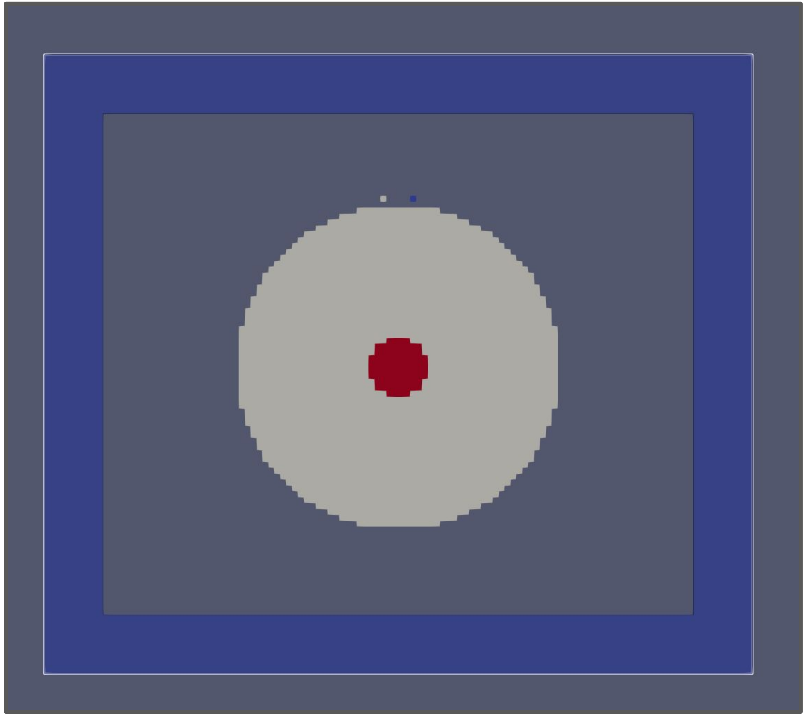
Overview

- Project Scope
- GPR Simulation
- Pole Classification AI
- Prototype Updates
- Future Work
- Summary

Project Scope

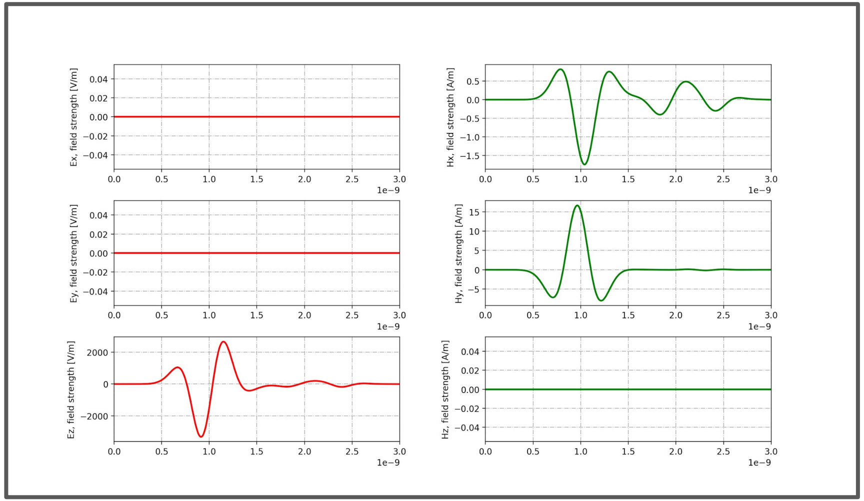
- Motivation:
  - Improve safety and reliability
  - Reduce resources needed to inspect poles
  - Increase inspection efficiency
- Goal:
  - Automate and simplify pole health inspection process

GPR Simulation

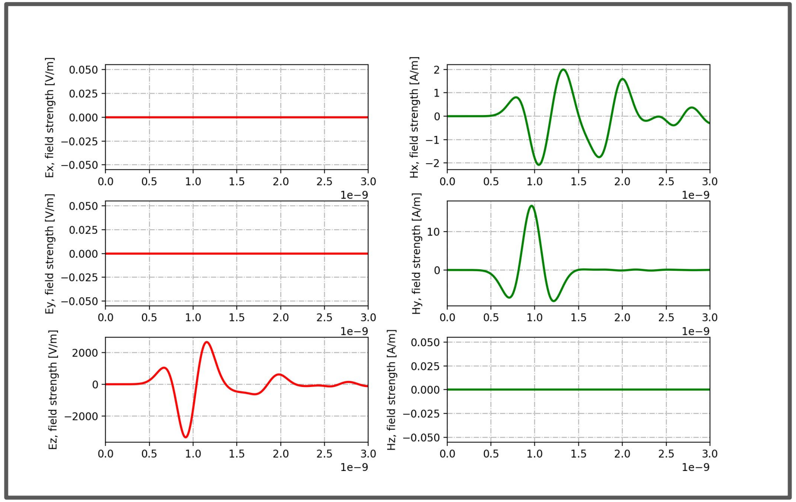


- gprMax
- Cross-section of utility pole
- Scaled down by 2

Simulation Results



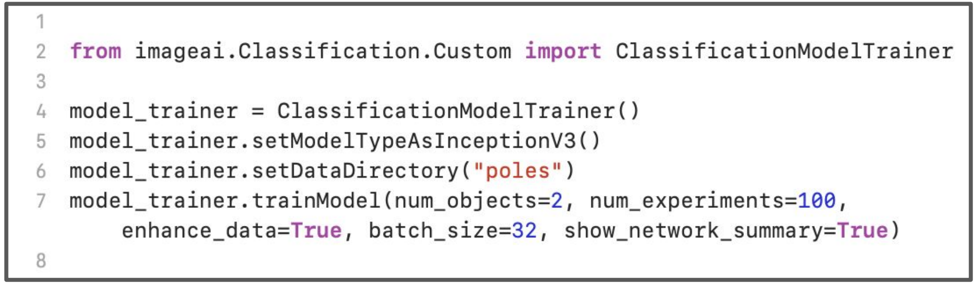
Control: Solid Pole (ϵ = 2)



Pole (ϵ = 2) with Void (ϵ = 20)

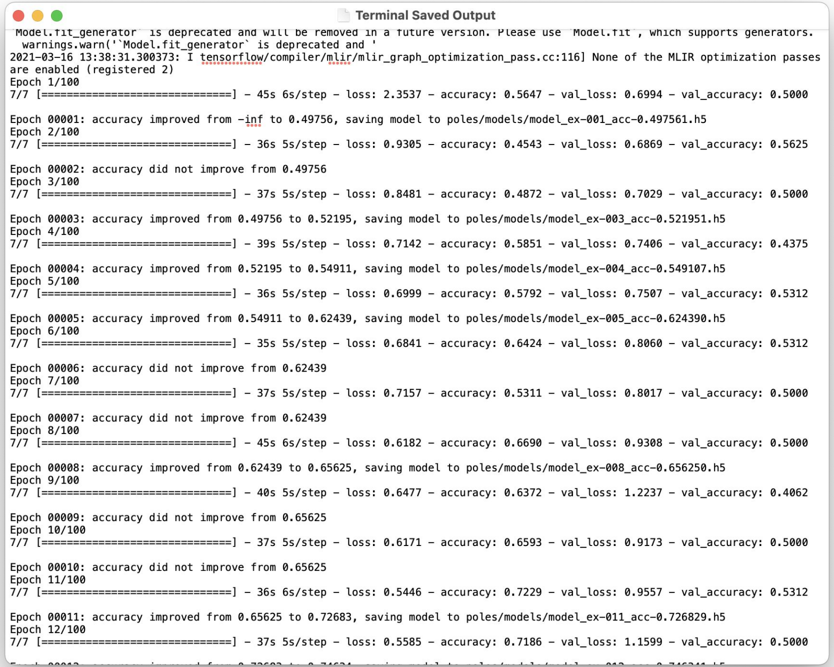
Image Classification

- ImageAI library
  - Python
  - Tensorflow
- Inception v3 algorithm
- 2 classes: good & bad
- 100 experiments
- Dataset
  - 200 training images
  - 40 test images



```

from imageai.Classification.Custom import ClassificationModelTrainer

model_trainer = ClassificationModelTrainer()
model_trainer.setModelTypeAsInceptionV3()
model_trainer.setDataDirectory("poles")
model_trainer.trainModel(num_objects=2, num_experiments=100,
    enhance_data=True, batch_size=32, show_network_summary=True)

```

Training

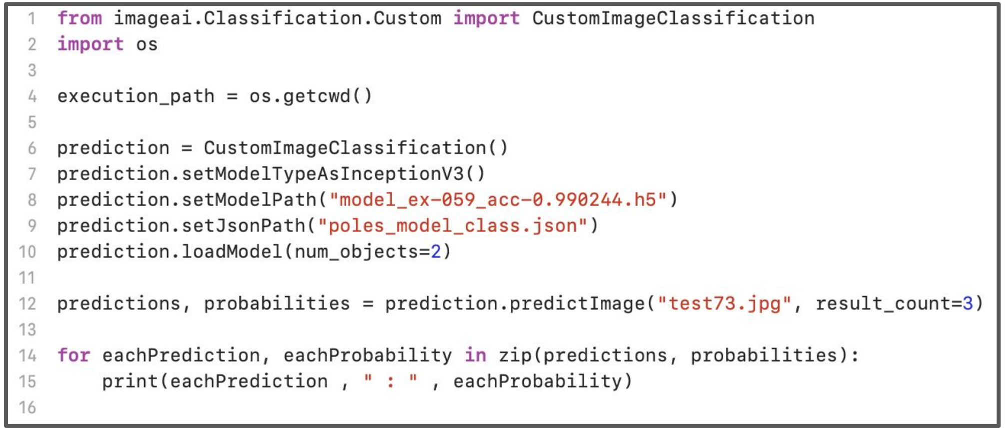


```
`Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
2021-03-16 13:38:31.300373: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes
are enabled (registered 2)
Epoch 1/100
7/7 [==============================] - 45s 6s/step - loss: 2.3537 - accuracy: 0.5647 - val_loss: 0.6994 - val_accuracy: 0.5000

Epoch 00001: accuracy improved from -inf to 0.49756, saving model to poles/models/model_ex-001_acc-0.497561.h5
Epoch 2/100
7/7 [==============================] - 36s 5s/step - loss: 0.9305 - accuracy: 0.4543 - val_loss: 0.6869 - val_accuracy: 0.5625

Epoch 00002: accuracy did not improve from 0.49756
Epoch 3/100
7/7 [==============================] - 37s 5s/step - loss: 0.8481 - accuracy: 0.4872 - val_loss: 0.7029 - val_accuracy: 0.5000

Epoch 00003: accuracy improved from 0.49756 to 0.52195, saving model to poles/models/model_ex-003_acc-0.521951.h5
Epoch 4/100
7/7 [==============================] - 39s 5s/step - loss: 0.7142 - accuracy: 0.5851 - val_loss: 0.7406 - val_accuracy: 0.4375

Epoch 00004: accuracy improved from 0.52195 to 0.54911, saving model to poles/models/model_ex-004_acc-0.549107.h5
Epoch 5/100
7/7 [==============================] - 36s 5s/step - loss: 0.6999 - accuracy: 0.5792 - val_loss: 0.7507 - val_accuracy: 0.5312

Epoch 00005: accuracy improved from 0.54911 to 0.62439, saving model to poles/models/model_ex-005_acc-0.624390.h5
Epoch 6/100
7/7 [==============================] - 35s 5s/step - loss: 0.6841 - accuracy: 0.6424 - val_loss: 0.8060 - val_accuracy: 0.5312

Epoch 00006: accuracy did not improve from 0.62439
Epoch 7/100
7/7 [==============================] - 37s 5s/step - loss: 0.7157 - accuracy: 0.5311 - val_loss: 0.8017 - val_accuracy: 0.5000

Epoch 00007: accuracy did not improve from 0.62439
Epoch 8/100
7/7 [==============================] - 45s 6s/step - loss: 0.6182 - accuracy: 0.6690 - val_loss: 0.9308 - val_accuracy: 0.5000

Epoch 00008: accuracy improved from 0.62439 to 0.65625, saving model to poles/models/model_ex-008_acc-0.656250.h5
Epoch 9/100
7/7 [==============================] - 40s 5s/step - loss: 0.6477 - accuracy: 0.6372 - val_loss: 1.2237 - val_accuracy: 0.4062

Epoch 00009: accuracy did not improve from 0.65625
Epoch 10/100
7/7 [==============================] - 37s 5s/step - loss: 0.6171 - accuracy: 0.6593 - val_loss: 0.9173 - val_accuracy: 0.5000

Epoch 00010: accuracy did not improve from 0.65625
Epoch 11/100
7/7 [==============================] - 36s 6s/step - loss: 0.5446 - accuracy: 0.7229 - val_loss: 0.9557 - val_accuracy: 0.5312

Epoch 00011: accuracy improved from 0.65625 to 0.72683, saving model to poles/models/model_ex-011_acc-0.726829.h5
Epoch 12/100
7/7 [==============================] - 37s 5s/step - loss: 0.5585 - accuracy: 0.7186 - val_loss: 1.1599 - val_accuracy: 0.5000

Epoch 00012: accuracy improved from 0.72683 to 0.74634, saving model to poles/models/model_ex-012_acc-0.746341.h5
```

Experiment Number	*Accuracy*
1	0.49756
3	0.52195
4	0.54911
5	0.62439
8	0.65625
11	0.72683
12	0.74634
14	0.76098
15	0.82439
16	0.85366
20	0.89268
21	0.91071
23	0.93659
24	0.95122
27	0.96585
29	0.97073
41	0.97561
43	0.98049
47	0.98661
59	0.99024

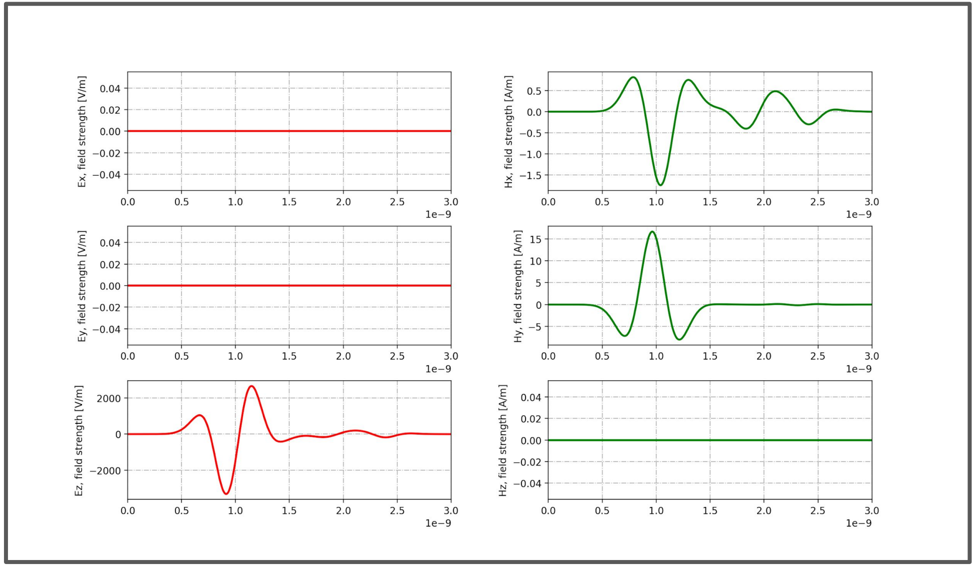
Testing

- Accurately determines good & bad poles
- Tested with a variety of scenarios



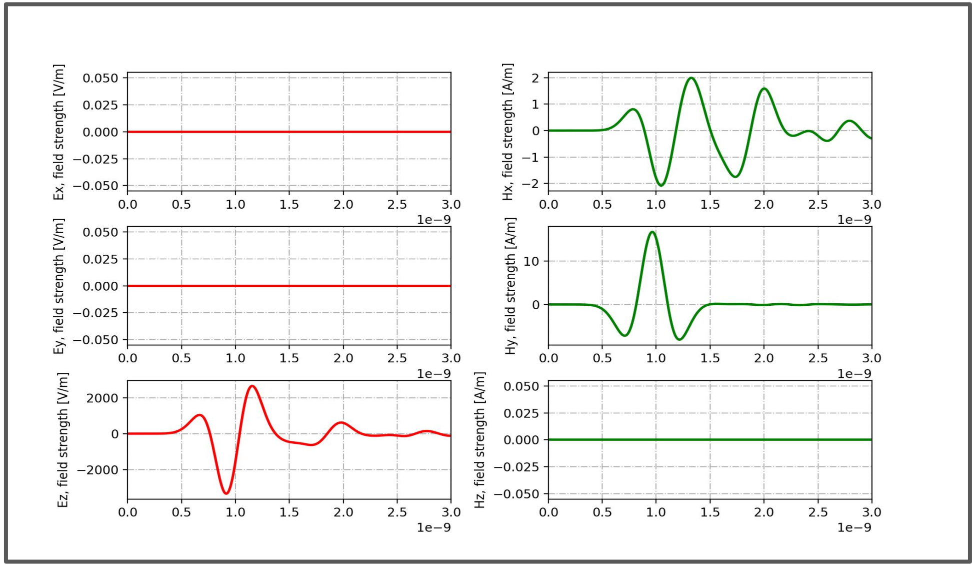
```python
from imageai.Classification.Custom import CustomImageClassification
import os

execution_path = os.getcwd()

prediction = CustomImageClassification()
prediction.setModelTypeAsInceptionV3()
prediction.setModelPath("model_ex-059_acc-0.990244.h5")
prediction.setJsonPath("poles_model_class.json")
prediction.loadModel(num_objects=2)

predictions, probabilities = prediction.predictImage("test73.jpg", result_count=3)

for eachPrediction, eachProbability in zip(predictions, probabilities):
    print(eachPrediction , " : " , eachProbability)

```

Testing



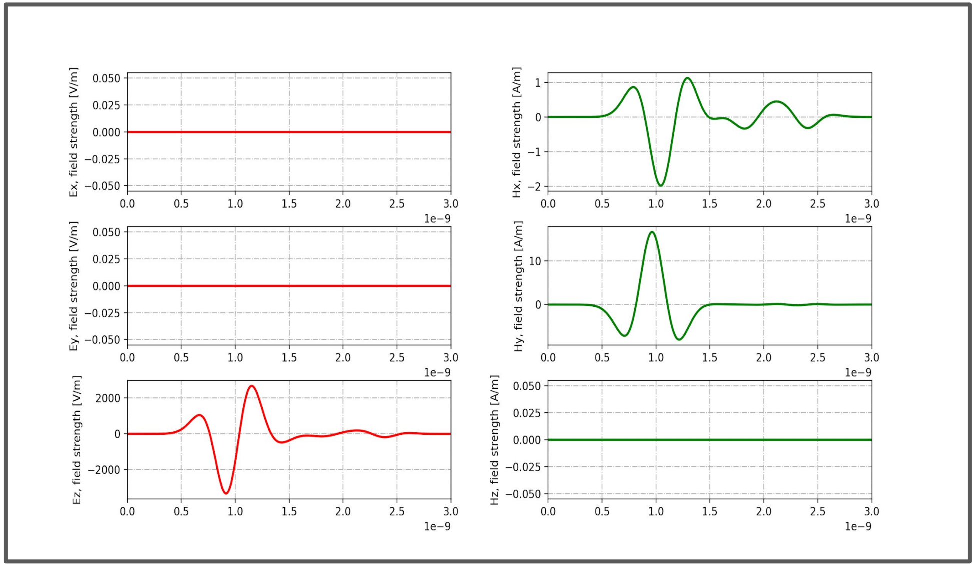
- Simulation Test 34
  - Solid Pole (ϵ = 2)
- Test Results
  - Good: 99.928087
  - Bad: 0.071916

Testing



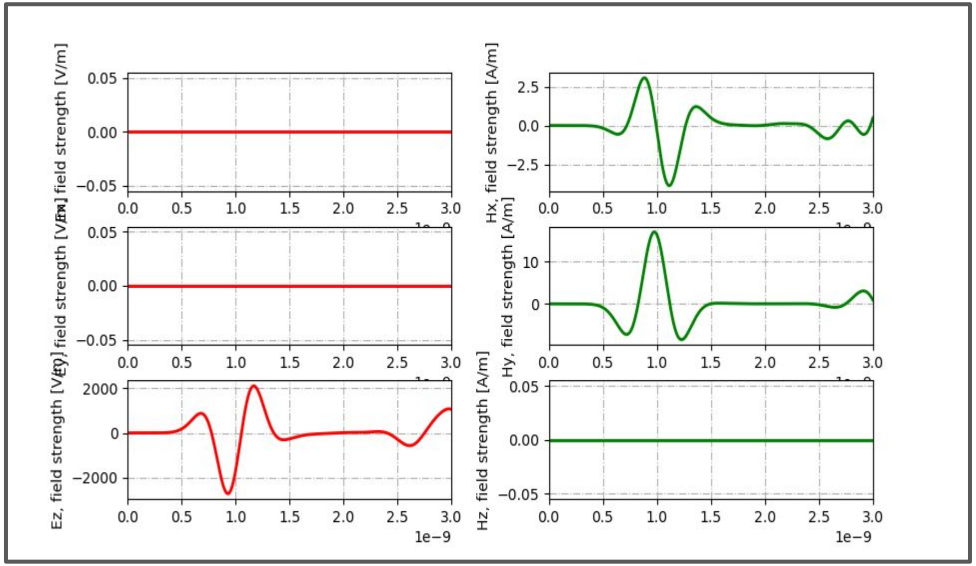
- Simulation Test 35
  - Pole (ϵ = 2) with centered void (ϵ = 20)
- Test Results
  - Good: 1.903430e-05
  - Bad: 99.999976

Testing



- Simulation Test 18
  - Pole (ϵ = 2) with offcentered air pocket (ϵ = 1)
- Test Results
  - Good: 2.127279
  - Bad: 97.872716

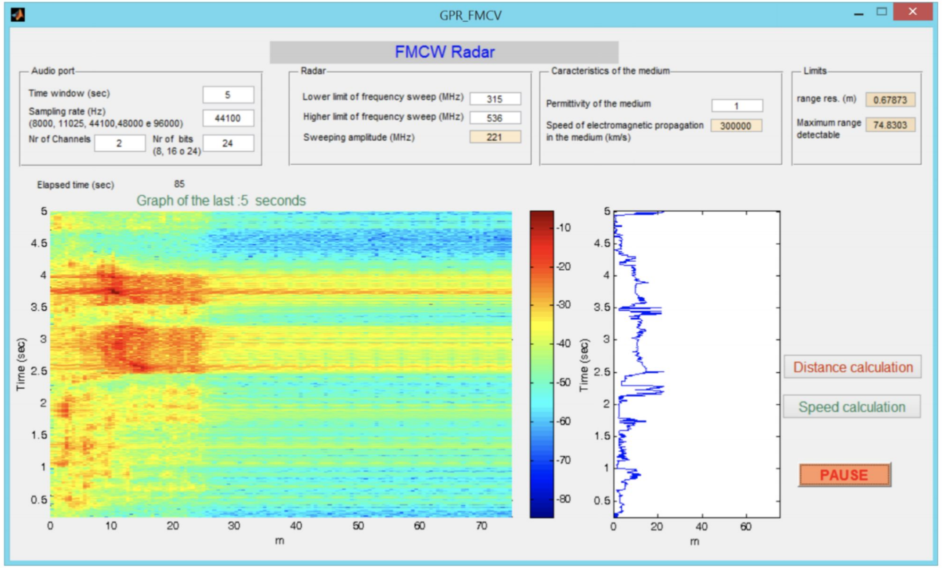
Testing



- Simulation Test 54
  - Solid Pole ($\epsilon = 6$)
- Test Results
  - Good: 98.765647
  - Bad: 1.234355

AI Adjustments

- Will need to change due to actual GPR results
- Retrain AI on good and bad sections of pole from FPL



GPR_FMCV

FMCW Radar

Audio port

Time window (sec): 5

Sampling rate (Hz) (8000, 11025, 44100,48000 e 96000): 44100

Nr of Channels: 2

Nr of bits (8, 16 o 24): 24

Radar

Lower limit of frequency sweep (MHz): 315

Higher limit of frequency sweep (MHz): 536

Sweeping amplitude (MHz): 221

Caracteristics of the medium

Permittivity of the medium: 1

Speed of electromagnetic propagation in the medium (km/s): 300000

Limits

range res. (m): 0.67873

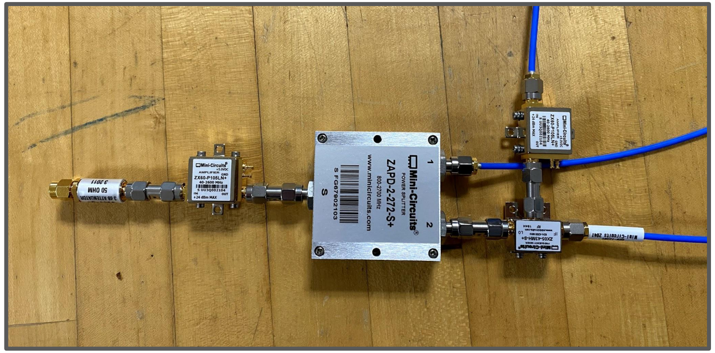
Maximum range detectable: 74.8303

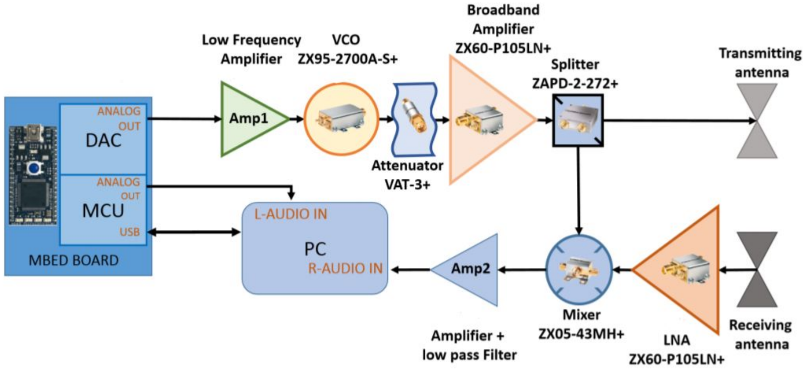
Elapsed time (sec): 85

Graph of the last :5 seconds

Prototype

- Major hardware components have been constructed
- Working on constructing subcomponents

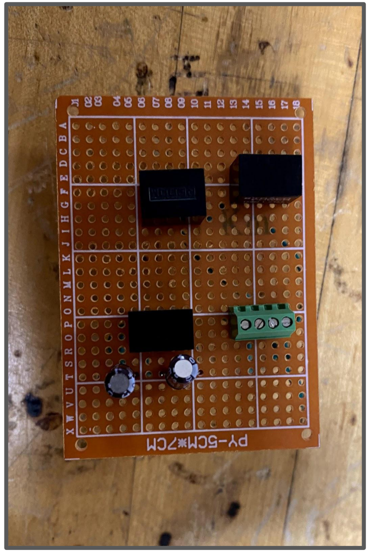


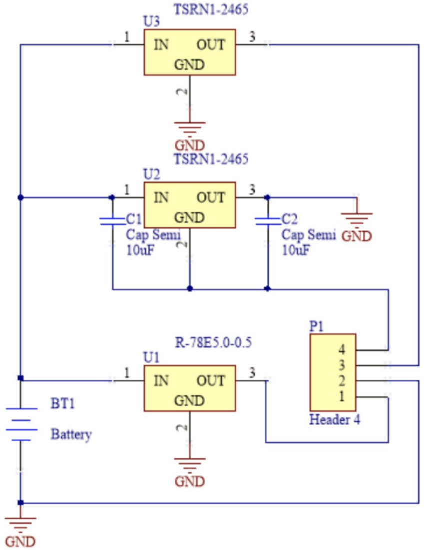


Hardware Subcomponents

DC to DC Converter

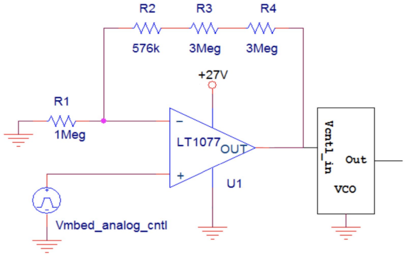
12V DC to four levels of output DC voltages: 5 V, ±6 V, GND

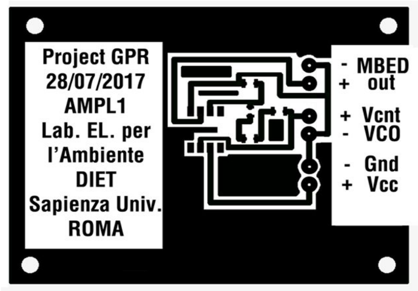




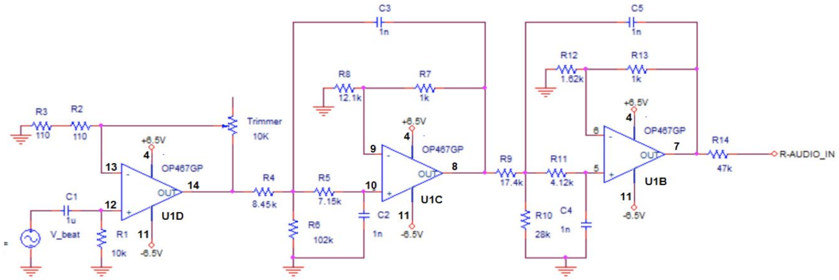
Hardware Subcomponents

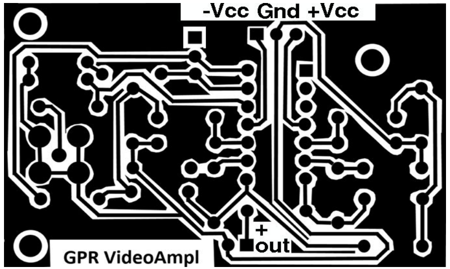
Amp1





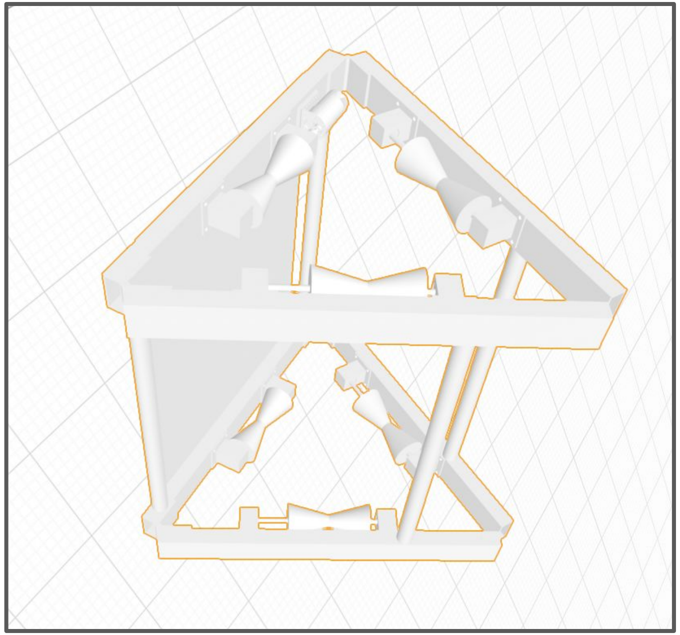
Amp2





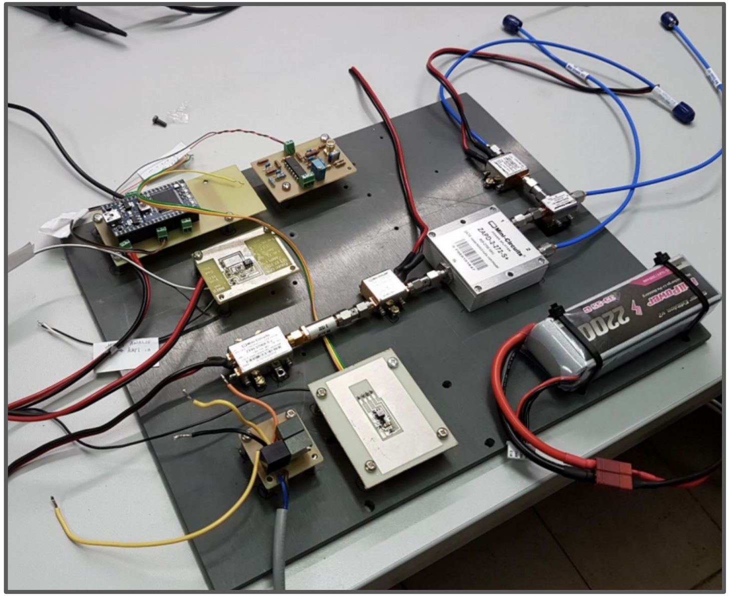
Robot Construction

- Pythagoras’ Collar
- Aluminum based
- GPR located on side plate



Future Work

- Construct complete prototype
- Add GPR component to ME robot
- Implement and adjust AI and GPR software
- Test and revise



Summary

- Developed a working image classification AI
- Building GPR subcomponents
- ME team has designed final robot prototype
- Start compiling all components together

Questions?

Title

- Text