A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one partially covering the green one.

Emulation of Land Survey System

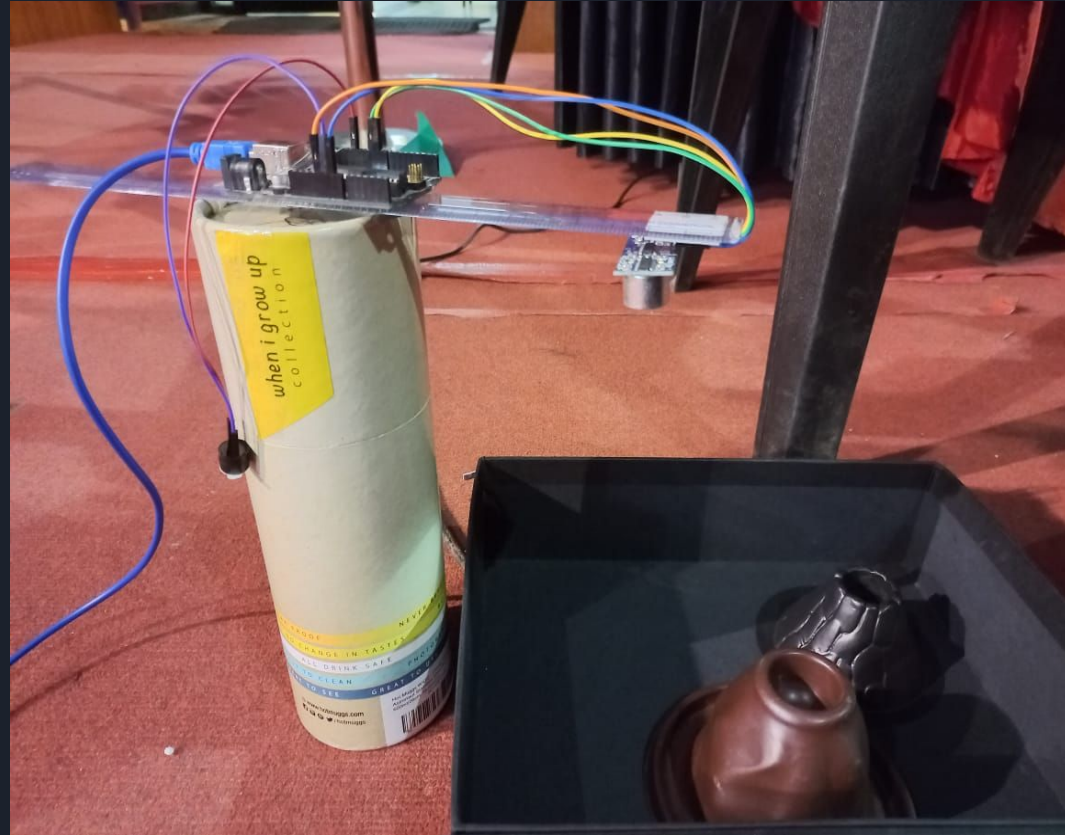
Team **Un-sensored**

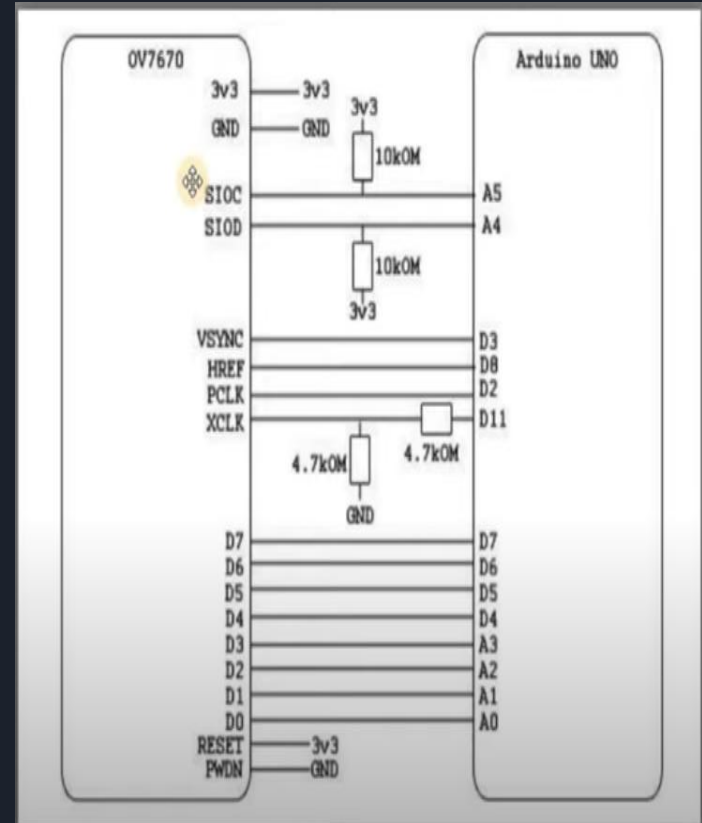
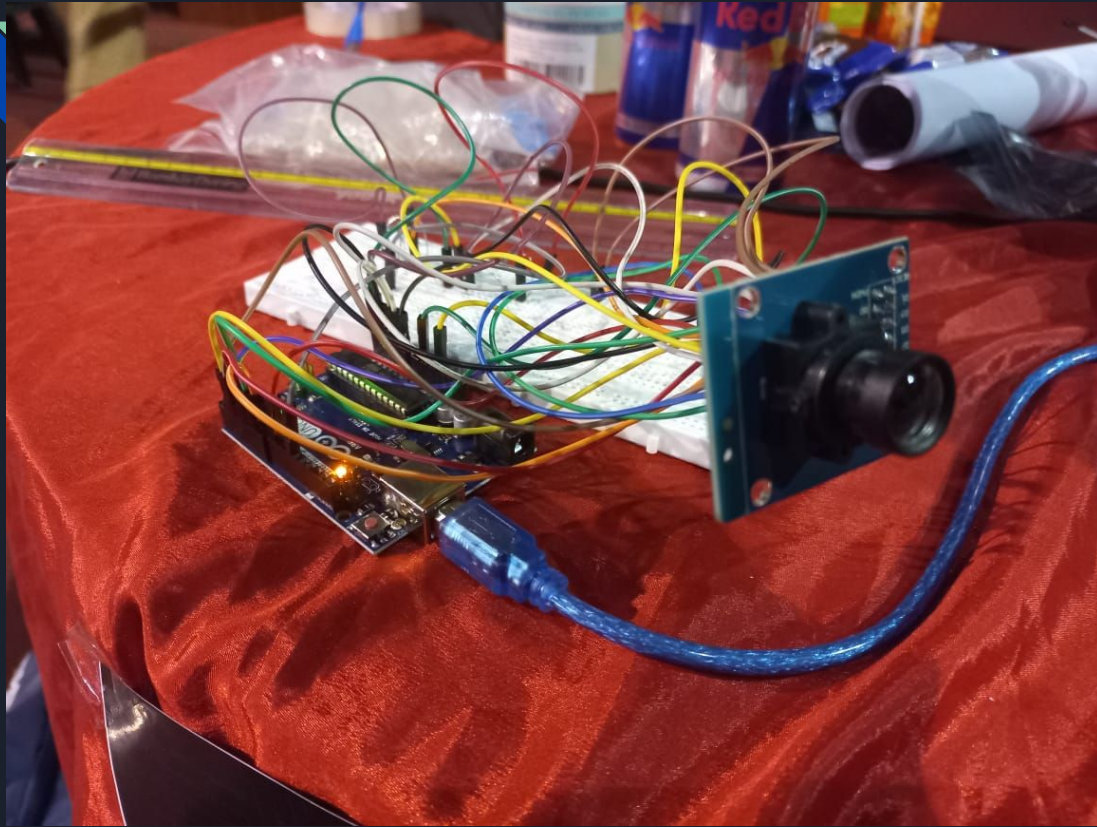


Abstract

- Farmers and real estate developers need an easy way to classify and survey land for sale and settling, to find unused land, or to check if the land is good or bad or flood prone.
- Farmers can use this surveyor to check if a land is fertile, or where the nearest water body is for agricultural purposes.
- Government officials can check for land that is barren for industrialisation.
- In the real estate sector this surveyor can be used to check if a piece of land is habitable or not.
- This surveyor captures images of land/water bodies and find the measurements, of the area.

The project





Prediction model

Storage

[Files](#) [Rules](#) [Usage](#)



Protect your Storage resources from abuse, such as billing fraud or phishing

[Configure App Check](#)



[gs://un-sensored-7f3d5.appspot.com](#)

[Upload file](#)



Name

Size

Type

Last modified



 6390_sat.jpg

2.52 MB

image/jpeg

Apr 14, 2022

```
import pyrebase
```

```
firebaseConfig = {  
    'apiKey': "AIzaSyBgKhRN5biUFhdOuuLspNCjVQTs8w4zHLk",  
    'authDomain': "un-sensored-7f3d5.firebaseio.com",  
    'databaseURL': "https://un-sensored-fe9de.firebaseio.com",  
    'projectId': "un-sensored-7f3d5",  
    'storageBucket': "un-sensored-7f3d5.appspot.com",  
    'messagingSenderId': "731735486291",  
    'appId': "1:731735486291:web:e478c750e77fcaae193ba6",  
    'measurementId': "G-QZ3557HTJ9",  
    'serviceAccount': "key.json"  
};
```

```
firebase_storage=pyrebase.initialize_app(firebaseConfig)
```

```
storage=firebase_storage.storage()
```

```
storage.child("6390_sat.jpg").download("6390_sat.jpg")#Calling image from firebase storage
```

```
result=cnn.predict("6390_sat.jpg")#running cnn predictor on said image
```

```
if result == 1:  
    print("crop")
```



```
from PIL import Image, ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True
cnn=tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3,activation='relu',input_shape=[64,64,3]))
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
cnn.fit(x = training_set,validation_data=test_set, epochs = 11)
```

Epoch 1/11

25/25 [=====] - 85s 3s/step - loss: 0.7350 - accuracy: 0.5611

Epoch 2/11

25/25 [=====] - 82s 3s/step - loss: 0.5172 - accuracy: 0.7619

Epoch 3/11

25/25 [=====] - 83s 3s/step - loss: 0.5340 - accuracy: 0.7130

Epoch 4/11

25/25 [=====] - 83s 3s/step - loss: 0.4295 - accuracy: 0.7915