Cropland mapping using random forest algorithm in Zambia

Yao-Ting Yao & Lester Carver

December 2022

Introduction

- Rapid expansion of crop fields sub-Saharan African countries
- Cropland mapping as a tool in food security research
- Study Area: Southern Province of Zambia
  - Largest area for commercial farmland

OBJECTIVE

Explore the performance of Random Forest Classification Models in predicting cropland in Zambia's Southern Province

Data

- **Administrative Boundaries**
- -OCHA Regional Office for Southern and Eastern Africa (ROSEA)
- **Imagery**

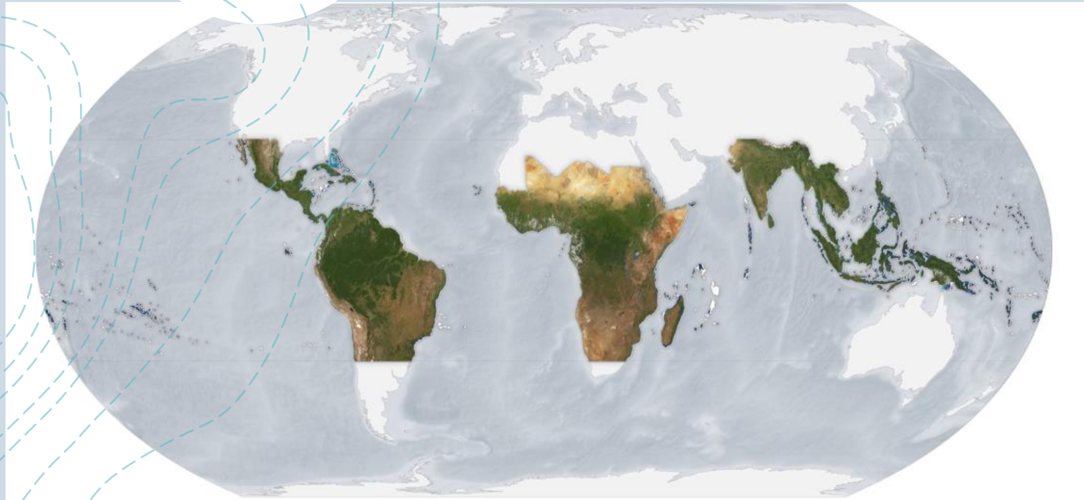
-USGS Landsat 8 Collection 2 Tier 1 Raw Scenes: 11 bands, 30m, 16days

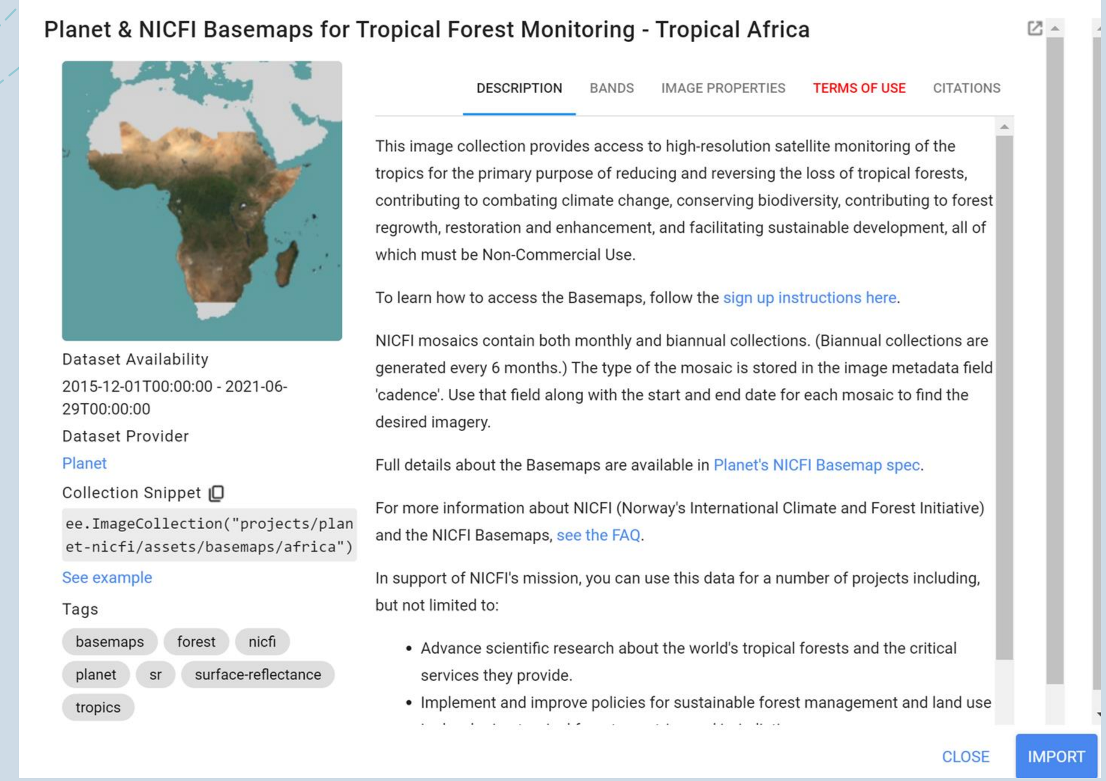
-Planet-Scope NICFI Tropical Basemaps

- **Sampling**

-UMD Global Cropland Dataset

-Label Dataset



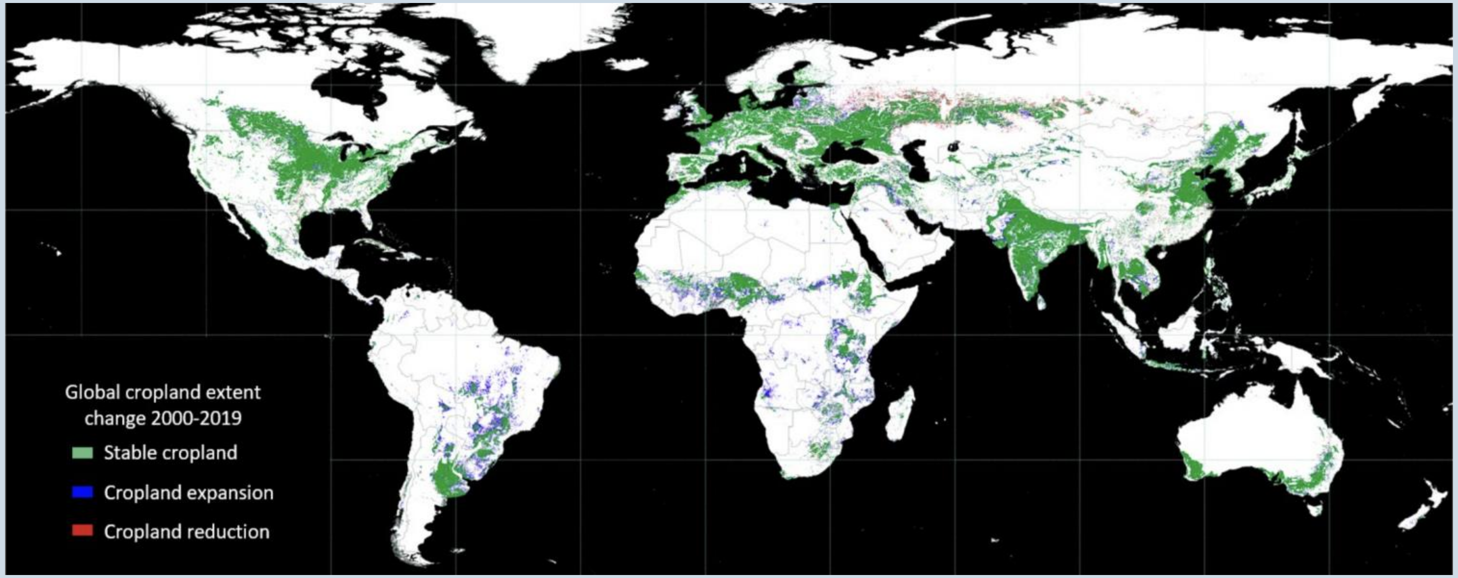


Planet-Scope NICFI Tropical Basemaps

- high spatial resolution (2.77m) : effective for classifying small cropland fields.
- 4-band multispectral (Blue, Green, Red, NIR)
- The revisit period is 1 day making Planet-Scope a perfect imagery to create large regional scale image mosaics for multiple periods during a single agricultural year.

UMD Global Cropland Dataset

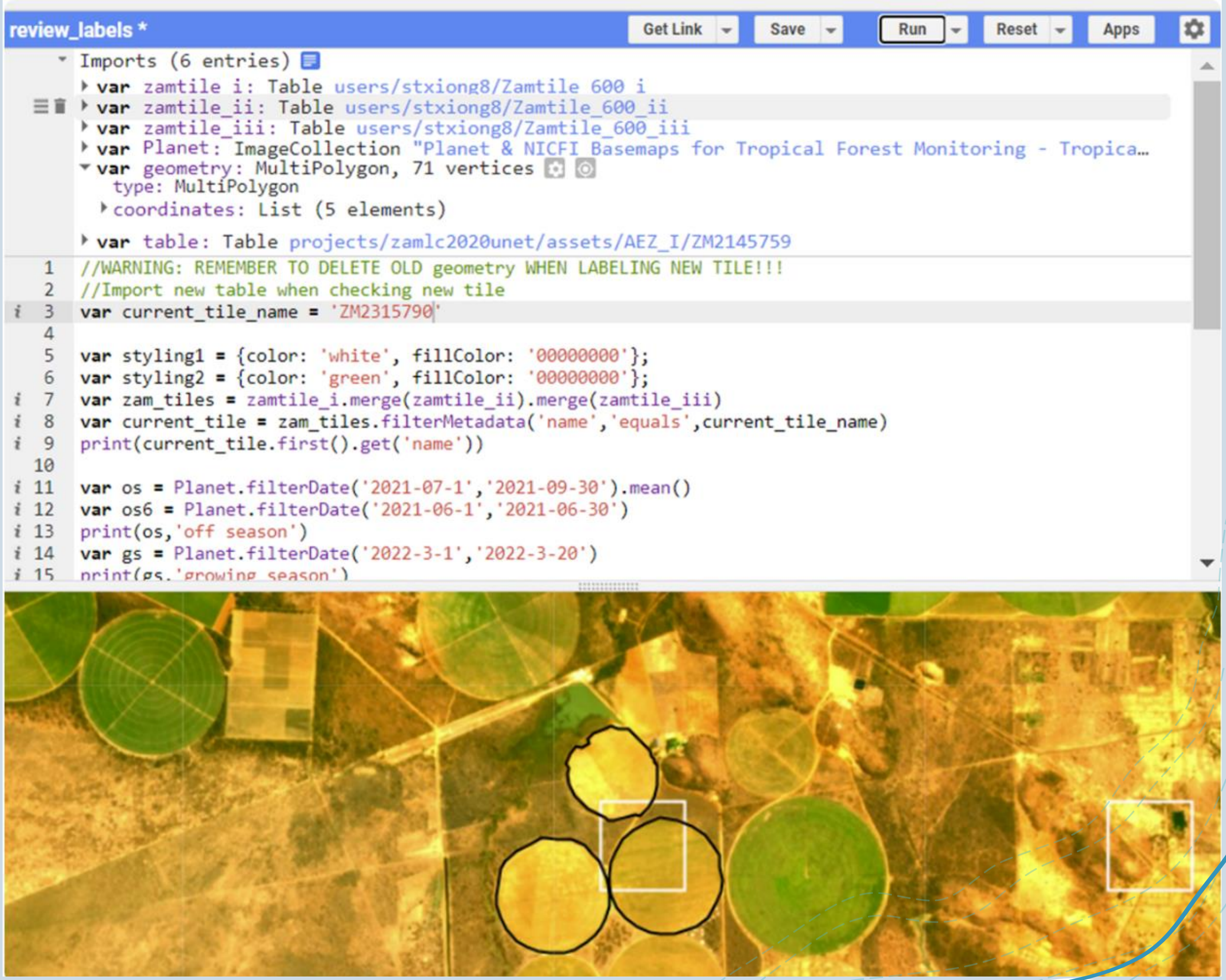
+ The Global Land Analysis and Discovery (GLAD) laboratory in the Department of Geographical Sciences at the University of Maryland.
+ The dataset represents a globally consistent cropland extent time-series at 30-m spatial resolution.
+ The cropland mapping was done using the consistently processed Landsat satellite data archive from 2000 to 2019.
+ We use Global_cropland_NE_2019 with crop and non-crop classes.



Global Land Analysis & Discovery

Label Dataset

- Created by Yao-Ting and Sitian through GEE platform based on Planet-Scope NICFI imagery using off-season imagery(2021-07-1- 2021-09-01)
- 600 labels. The size of the grid is 200 by 200, which is randomly selected in Zambia agriculture zone one (southern part of Zambia).
- We used 304 labels located in Southern Province of Zambia as one of our sampling source.



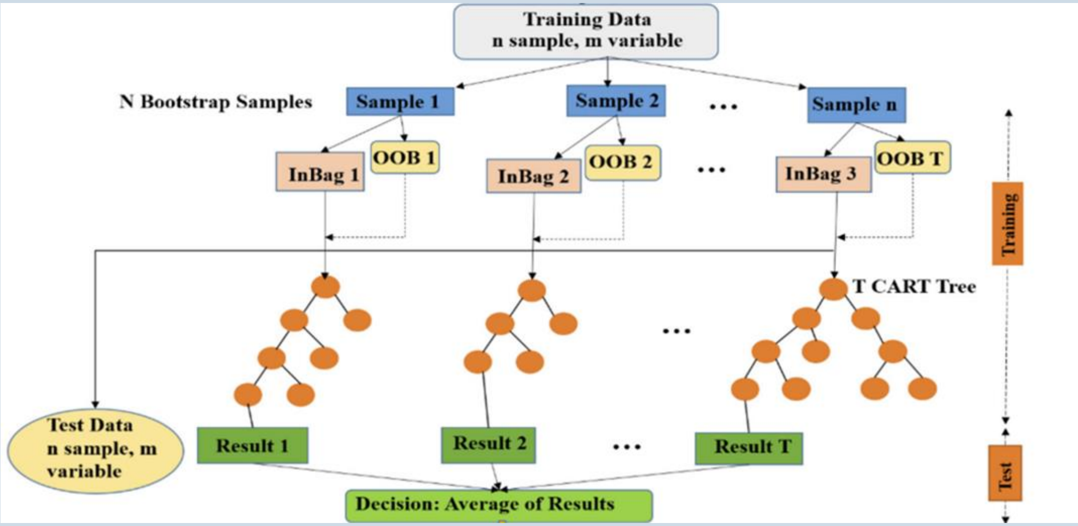
Methodology

- Random Forest (RF) in Google Earth Engine
- A random forest (RF) classifier is an ensemble classifier that produces multiple decision trees, using a randomly selected subset of training samples and variables (Breiman, 2001).
- GEE **ee.Classifier.smileRandomForest** module

- 1000 sample points: 80% In Bag for training; 20% Out of Bag(OOB) for validation.

- We trained the model with tree number of 100

- The variables are R, G, B, NIR, NDVI, and GCVI, totally of six variables.



Home > Products > Google Earth Engine > Reference

Was this helpful?

ee.Classifier.smileRandomForest

Send feedback

Creates an empty Random Forest classifier.

Usage	Returns
`ee.Classifier.smileRandomForest(numberOfTrees, variablesPerSplit, minLeafPopulation, bagFraction, maxNodes, seed)`	Classifier

Argument	Type	Details
`numberOfTrees`	Integer	The number of decision trees to create.
`variablesPerSplit`	*Integer, default: null*	*The number of variables per split. If unspecified, uses the square root of the number of variables.*
`minLeafPopulation`	*Integer, default: 1*	*Only create nodes whose training set contains at least this many points.*
`bagFraction`	*Float, default: 0.5*	*The fraction of input to bag per tree.*
`maxNodes`	*Integer, default: null*	*The maximum number of leaf nodes in each tree. If unspecified, defaults to no limit.*
`seed`	*Integer, default: 0*	*The randomization seed.*

Process

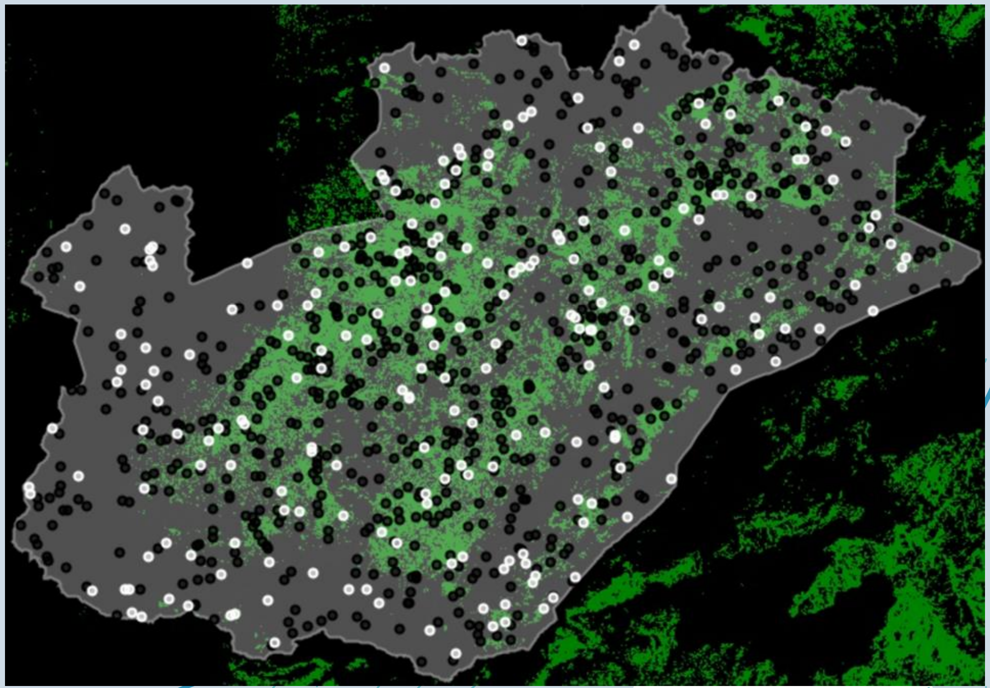
Create 2 sample points feature collection	Create 2 image composite with 6 variables band	Create 4 RF classifier: train a 100-tree and validation.	Create variable importance charts	Calculate total area of crop and non-crop class

Create 1000 sample points feature collection

1. UMD Global Cropland Dataset: split training and validation

```
//PART 3 - SETTING RANDOM SAMPLES
// Global Cropland data from University of Maryland's Global Land & Discovery Lab
var dat = ee.ImageCollection('users/potapovpeter/Global_cropland_2019').toList(10).get(2
var dat = ee.Image(dat)

// Remap the land cover class values to a 0-based sequential series.
var classValues = [0, 1];
var remapValues = ee.List.sequence(0, 1);
var label = 'dat';
dat = dat.remap(classValues, remapValues).rename(label).toByte();
print(dat)

// Add land cover as a band of the reflectance image and sample 200 pixels at
// 50 m scale from each land cover class within a region of interest.
// Use digitize tool to draw a polygon for sample region
var sample = img.addBands(dat).stratifiedSample({
  numPoints: 500,
  classBand: label,
  region: south_zambia,
  scale: 50,
  geometries: true
});

// Add a random value field to the sample and use it to approximately split 80%
// of the features into a training set and 20% into a validation set.
sample = sample.randomColumn();
var trainingSample = sample.filter('random <= 0.8');
var validationSample = sample.filter('random > 0.8');
print(sample)
```


FeatureCollection (1000 elements,… JSON
type: FeatureCollection
columns: Object (8 properties)
features: List (1000 elements)
0: Feature 0 (Point, 8 properties)
type: Feature
id: 0
geometry: Point (26.23, -17.54)
properties: Object (8 propertie…
B2: 0.11291389167308807
B3: 0.10536516457796097
B4: 0.12294043600559235
B5: 0.25943151116371155
GCVI: 1.4622132778167725
NDVI: 0.35695892572402954
dat: 0
random: 0.04581316436487448
1: Feature 1 (Point, 8 properties)
2: Feature 2 (Point, 8 properties)
3: Feature 3 (Point, 8 properties)
4: Feature 4 (Point, 8 properties)
5: Feature 5 (Point, 8 properties)
6: Feature 6 (Point, 8 properties)
7: Feature 7 (Point, 8 properties)
8: Feature 8 (Point, 8 properties)
9: Feature 9 (Point, 8 properties)

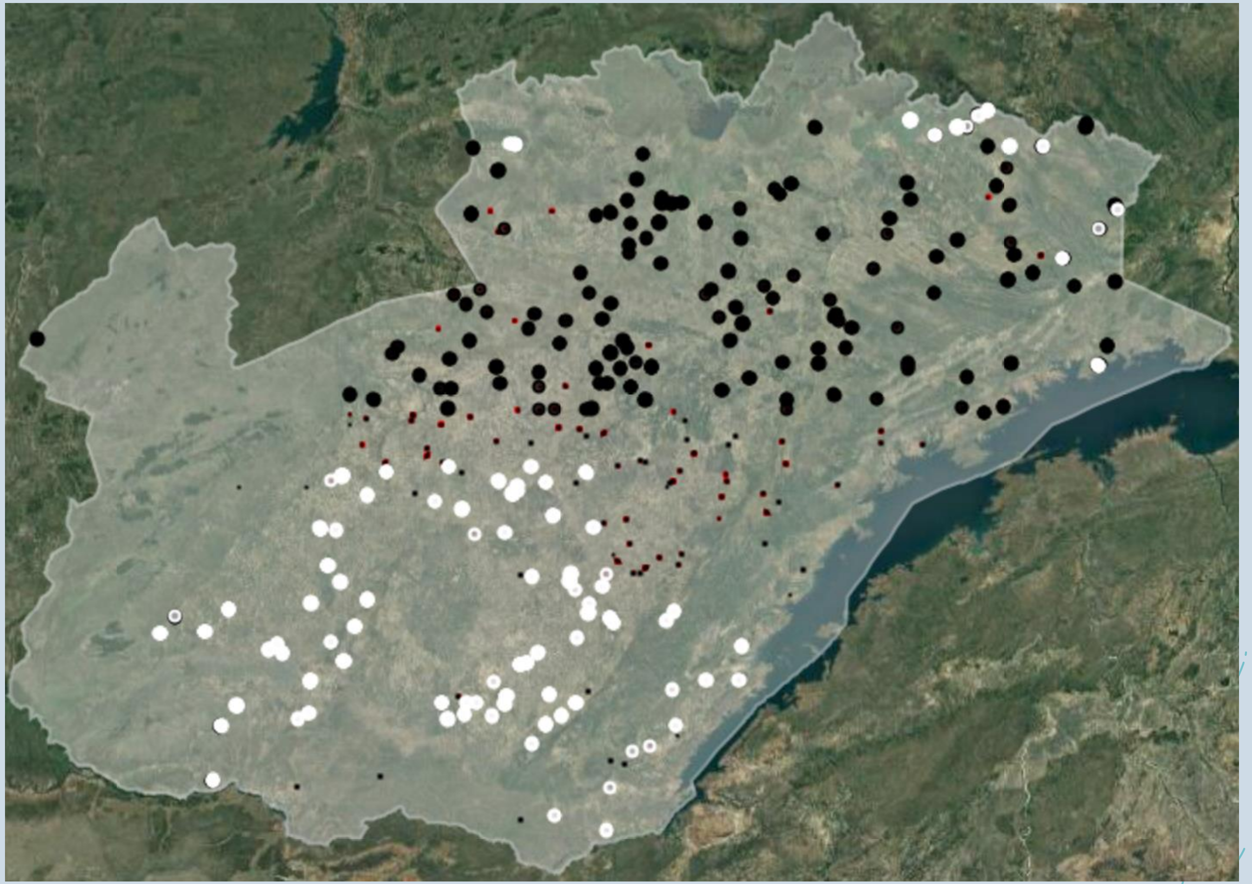


2. Label Dataset:

- Create field and non-field feature collection

- Use ee.Image.sampleRegions module

```
//Non_field sample, select 500 point
var non_field_sample = img.sampleRegions({
  collection: non_field,
  scale: 250,
  geometries: true
});
var non_field_sample = non_field_sample.toList(500)
var non_field_sample = ee.FeatureCollection(non_field_sample)
print(non_field_sample)
Map.addLayer(non_field_sample,{},'non_field_sample')

//Field sample, select 500 point
var field = ee.FeatureCollection(ee.Feature(field_in_grid, {p:1}));
var field_sample = img.sampleRegions({
  collection: field,
  scale: 150,
  geometries: true
});
var field_sample = field_sample.toList(500)
var field_sample = ee.FeatureCollection(field_sample)
print(field_sample)
Map.addLayer(field_sample,{color:'white'},'field_sample')

//merge 2 sample feature collection and export to asset
var sample = field_sample.merge(non_field_sample)
Export.table.toAsset({
  collection:sample,
  description:'sample',
  assetId: 'sample'
  });
```



Create an image composite with 6 variables bands

- Create Landsat 8 and Planet image composite
- Use ee.Algorithm.Landsat.simpleComposite() to create composite
- NDVI band: B5 is NIR, B4 is Red, using normalizedDifference()
- GCVI band: NIR/G-1
- Use ee.image.select() choose R,G,B,NIR
- ee.image.addBand to add NDVI and GCVI

```
//PART 2 - SETTING IMAGERY BANDS
//This script is tested using Landsat 8 Collection 2 Tier 1 imagery
//Computes a Landsat TOA composite to remove cloud cover
var composite = ee.Algorithms.Landsat.simpleComposite({
  collection: Landsat8.filterDate('2021-01-01', '2021-12-31'),
  asFloat: true
})
//Calculate NDVI
var ndvi = composite.normalizedDifference(['B5', 'B4']).rename("NDVI")
//Calculate GCVI with Map Algebra
var nir = composite.select('B5'); //Sub 'B5' with 'NIR'
var green = composite.select('B3');//Sub 'B3' with 'G'
var gcvi = (nir.divide(green)).subtract(1).rename("GCVI");
//Convert image collection to image
var image = ee.Image(composite)
//Add ndvi and gcvi into bands, totally 10 bands
var img = image.select(['B2', 'B3', 'B4', 'B5']).addBands(ndvi).addBands(gcvi);
```

Random Forest

- Train a 100-tree random forest classifier from the training sample.
- Get a train/validation error matrix and training overall accuracy

```
//PART4 - RANDOM FOREST
// Train a 100-tree random forest classifier from the training sample.
var trainedClassifier = ee.Classifier.smileRandomForest(100).train({
  features: trainingSample,
  classProperty: label,
  inputProperties: img.bandNames()
});
// Get information about the trained classifier.
var dict = trainedClassifier.explain();
// Get a confusion matrix and overall accuracy for the training sample.
var trainAccuracy = trainedClassifier.confusionMatrix();
// Get a confusion matrix and overall accuracy for the validation sample.
validationSample = validationSample.classify(trainedClassifier);
var validationAccuracy = validationSample.errorMatrix(label, 'classification');
// Classify the reflectance image from the trained classifier.
var imgClassified = img.classify(trainedClassifier);
```

Create variable importance charts

```
//variable importance chart
var variable_importance = ee.Feature(null, ee.Dictionary(dict).get('importance'));
var chart =
  ui.Chart.feature.byProperty(variable_importance)
    .setChartType('ColumnChart')
    .setOptions({
      title: 'Random Forest Variable Importance',
      legend: {position: 'none'},
      hAxis: {title: 'Bands'},
      vAxis: {title: 'Importance'}
    });
print(chart);
```

Calculate total area of crop and non-crop class

- ee.Image.pixelArea() to calculate total area
- ee.Image.reduceRegion() to sum the area base on group classification

Object (1 property) JSON
groups: List (2 elements)
0: Object (2 properties)
classification: 0
sum: 53772763953.90855
1: Object (2 properties)
classification: 1
sum: 13780460007.51782

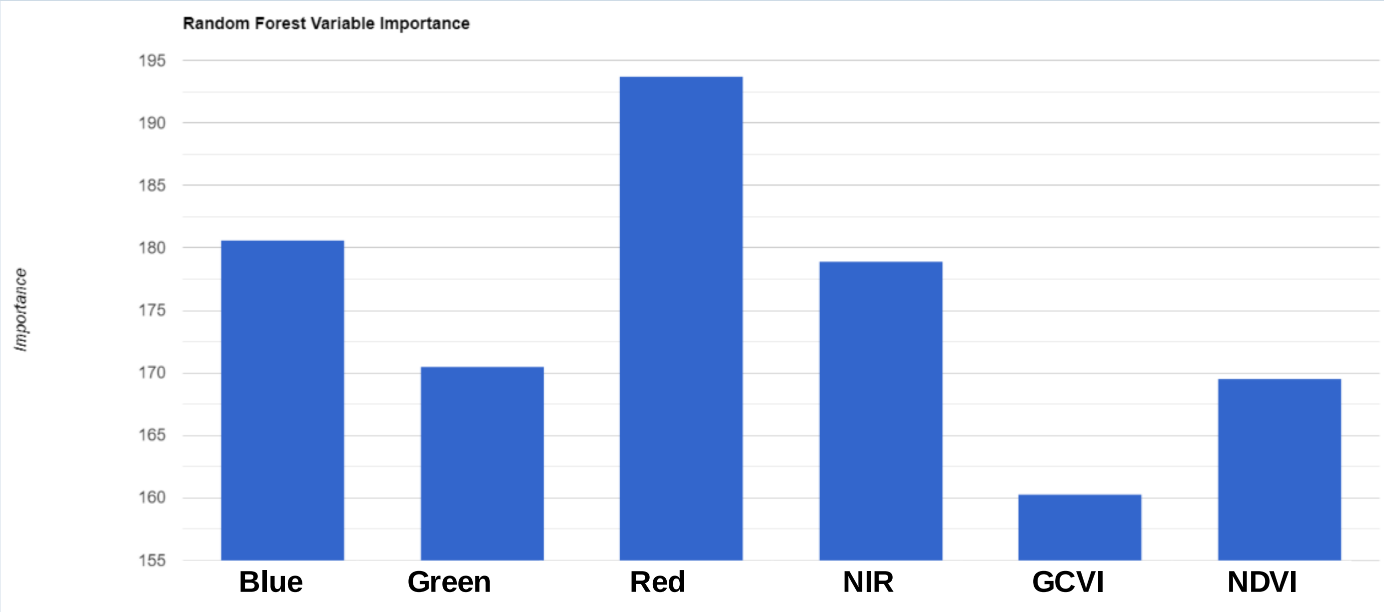

```
//Part5 Visulization of result:
//Clip to studay area
var imgClassified_SA =imgClassified.clipToCollection(south_zambia);
Map.addLayer(imgClassified_SA,classVis, 'imgClassified_SA');

//Area of field and non-field class
var areaImage = ee.Image.pixelArea().addBands(
    imgClassified_SA)
var areas = areaImage.reduceRegion({
    reducer: ee.Reducer.sum().group({
    groupField: 1,
    groupName: 'classification',
  }),
  geometry: south_zambia1,
  scale: 500,
  maxPixels: 1e10
  });
print(areas)
```

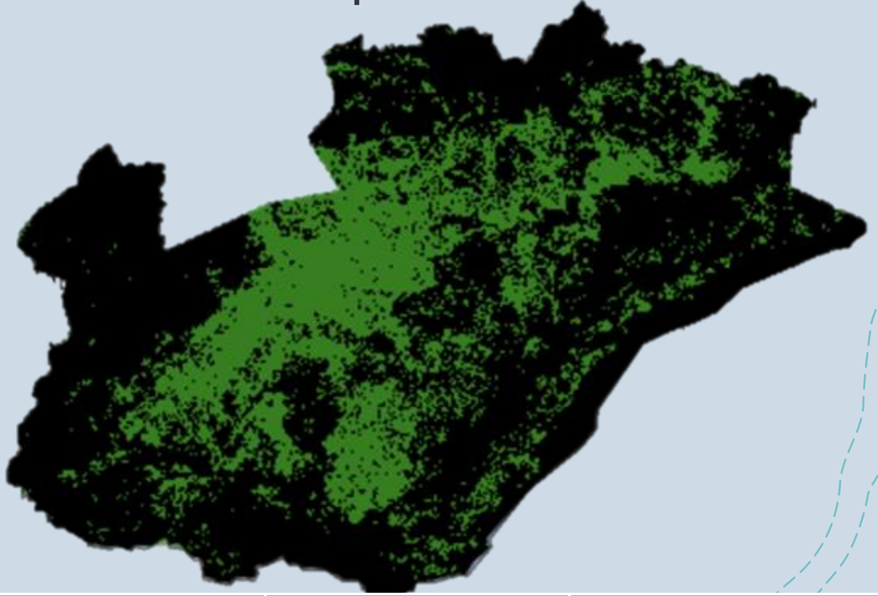
Results: LandSat 8 – UMD Global Cropland Sample

+ Training Accuracy: .9874
+ Validation Accuracy: .7921
+ Estimated Cropland: 20438.8577 km^2

	Non-Crop	Crop
Non-Crop	387	6
Crop	4	401

	Non-Crop	Crop
Non-Crop	83	24
Crop	18	77





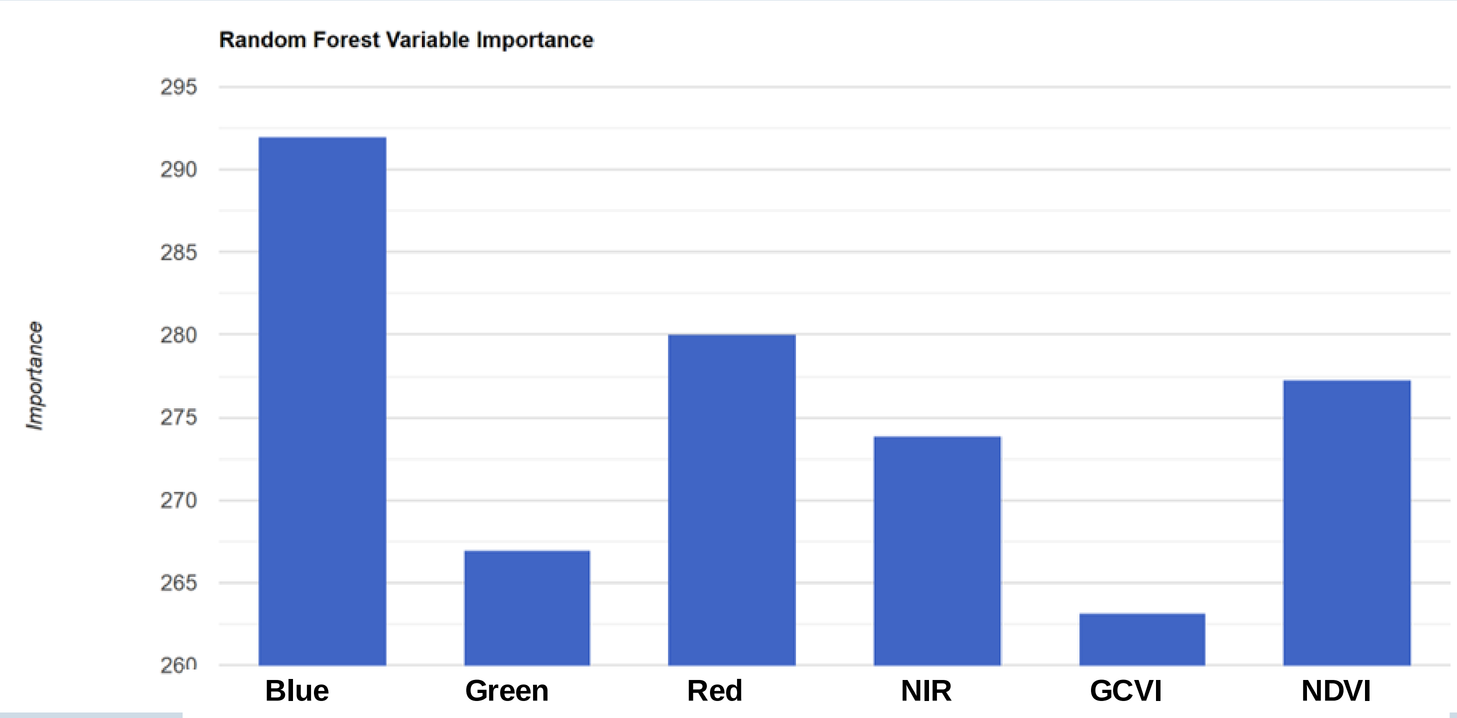
Satellite Imagery	UMD Imagery	Model Prediction
		

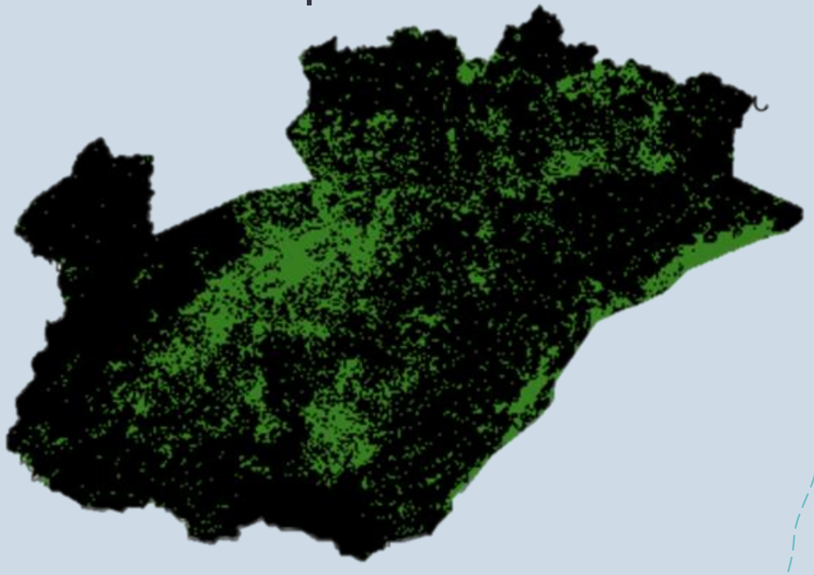
Results: LandSat 8 – Label Sample

- Training Accuracy: .9852
- Validation Accuracy: .7632
- Estimated Cropland: 13780.4600 km^2

	Non-Crop	Crop
Non-Crop	409	6
Crop	6	389

	Non-Crop	Crop
Non-Crop	70	15
Crop	30	75





Satellite Imagery	UMD Imagery	Model Prediction

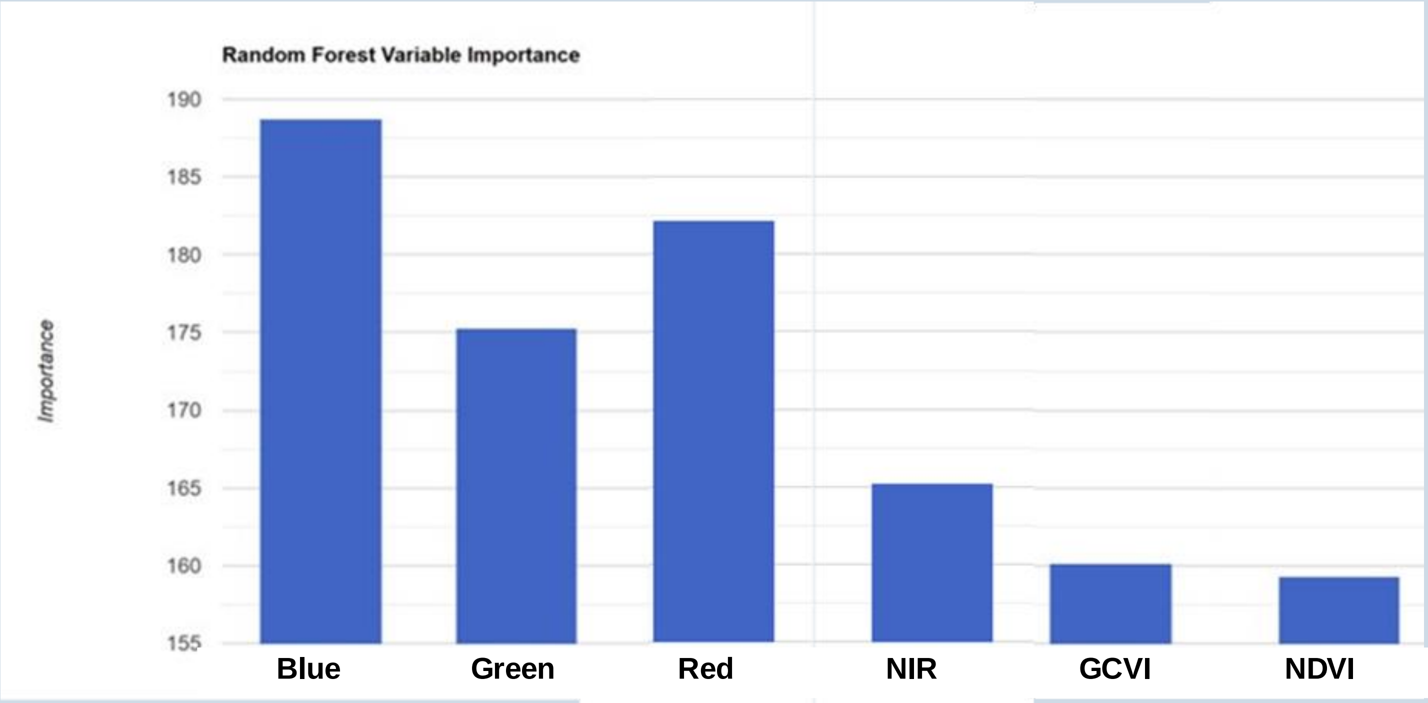


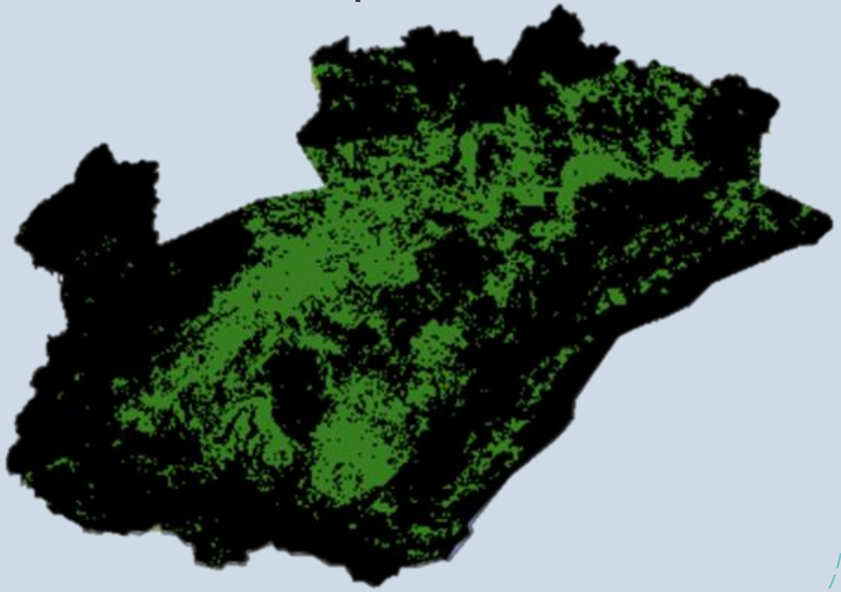
Results: Planet Scope – UMD Global Cropland Sample

+ Training Accuracy: .9825
+ Validation Accuracy: .8894
+ Estimated Cropland: 17878.7731 km^2

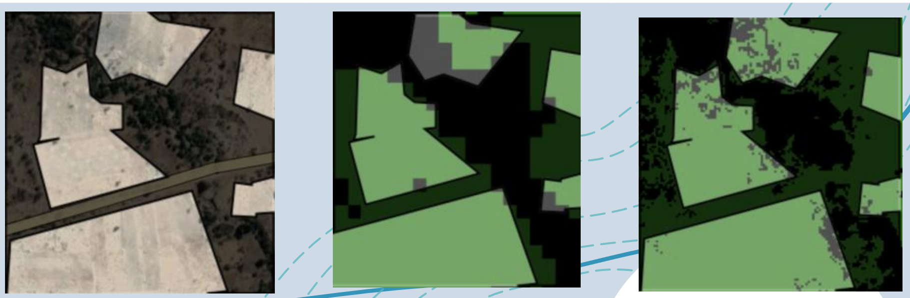
	Non-Crop	Crop
Non-Crop	403	6
Crop	8	384

	Non-Crop	Crop
Non-Crop	81	10
Crop	12	96





Satellite Imagery	UMD Imagery	Model Prediction



Results: Planet Scope – Label Sample

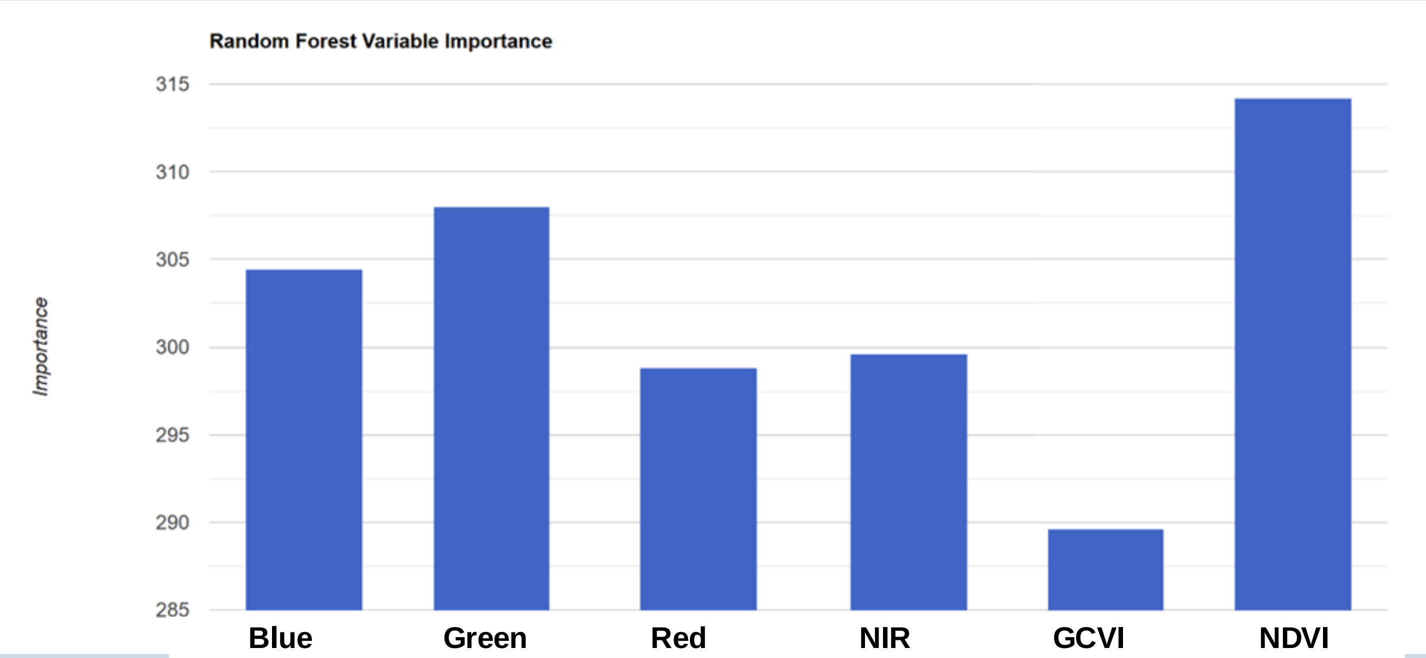
- Training Accuracy: .9902

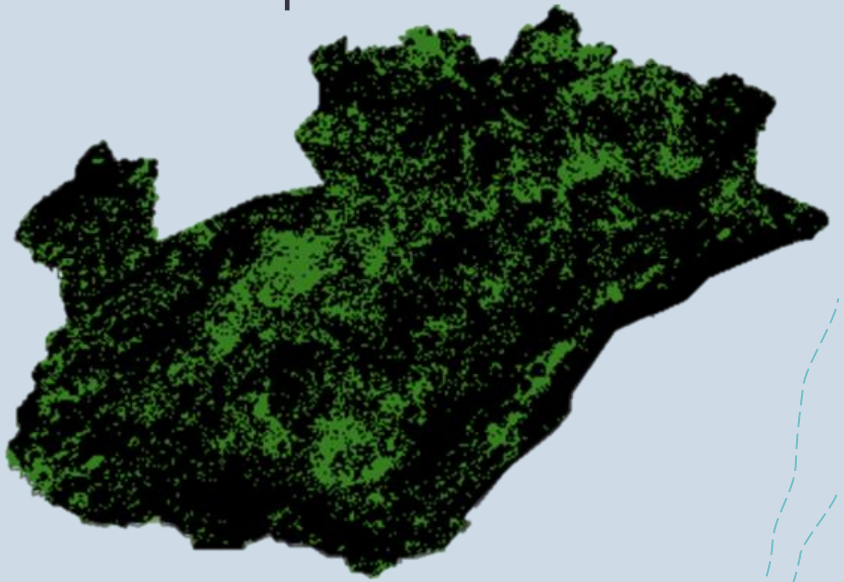
	Non-Crop	Crop
Non-Crop	403	5
Crop	3	403

- Validation Accuracy: .7097

	Non-Crop	Crop
Non-Crop	67	25
Crop	29	65

- Estimated Cropland: 17162.0970 km^2





Satellite Imagery	UMD Imagery	Model Prediction



Conclusions

+ All of the models performed well with .98+ training accuracy
+ Planet Scope imagery with the Label sampling data has the highest training accuracy, but the lowest validation accuracy, which suggests this model overfitted to the training data
+ Planet Scope imagery with the UMD Global Cropland sampling data has the lowest training accuracy, but the highest validation accuracy and the lowest difference between training and validation accuracy, making it overall the best model.

Conclusions

- Planet Scope imagery has a better spatial resolution (2.77m) than Landsat imagery (30m). So, the cropland boundary has more details in the map created with Planet Scope imagery.
- The labeled dataset, 304 labels with a 200 by 200 dimension grid, is far smaller than the Global Cropland dataset which includes less context of cropland when we used to train the model. As a result, the sample points we created from the Global Cropland dataset have a better result for prediction.

References

- *Zambia. IFAD. (n.d.). Retrieved November 13, 2022, from https://www.ifad.org/en/web/operations/w/country/zambia*
- *Breiman L (2001) Random forests. Mach Learn 45(1):5–32*
- *Debats, S. R., Luo, D., Estes, L. D., Fuchs, T. J., and Caylor, K. K. (2016). A Generalized Computer Vision Approach to Mapping Crop fields in Heterogeneous Agricultural Landscapes. Remote Sensing Environ. 179, 210–221. doi:10.1016/j.rse.2016.03.010*
- *Estes, L., et al. (2022). High Resolution, Annual Maps of Field Boundaries for Smallholder-Dominated Croplands at National Scales. Front. Artif. Intell.*
- *Sitian, X., et al. (2022). Probabilistic Tracking of Annual Cropland Changes over Large, Complex Agricultural Landscapes Using Google Earth Engine, Remote Sensing, 14, 4896.*