

For Kharif crop:

```
function maskS2clouds(image) {
  var qa = image.select('QA60');
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
    .and(qa.bitwiseAnd(cirrusBitMask).eq(0));
  return image.updateMask(mask).divide(10000);
}
var dt = table.filter(ee.Filter.eq('DISTNAME', 'CHURU'));
var coll1 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-04-01', '2020-04-15')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
var coll2 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-04-16', '2020-04-30')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
var coll3 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-05-01', '2020-05-15')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
var coll4 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-05-16', '2020-05-31')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
var coll5 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-06-01', '2020-06-15')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
var coll6 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
```

```

        .filterDate('2020-06-16','2020-06-30')
        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
            .map(maskS2clouds);
        .filterBounds(dt);
var coll7 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
        .filterDate('2020-07-01','2020-07-15')
        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
            .map(maskS2clouds);
        .filterBounds(dt);
var coll8 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
        .filterDate('2020-07-16','2020-07-31')
        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
            .map(maskS2clouds);
        .filterBounds(dt);
var coll9 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
        .filterDate('2020-08-01','2020-08-15')
        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
            .map(maskS2clouds);
        .filterBounds(dt);
var coll10 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
        .filterDate('2020-08-16','2020-08-31')
        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
            .map(maskS2clouds);
        .filterBounds(dt);
var coll11 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
        .filterDate('2020-09-01','2020-09-15')
        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
            .map(maskS2clouds);
        .filterBounds(dt);
var coll12 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
        .filterDate('2020-09-16','2020-09-30')
        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))

```

```

        .map(maskS2clouds);

        .filterBounds(dt);

print(coll1);

var addNDVI = function(img){
    var ndvi = img.normalizedDifference(['B8','B4']).rename('NDVI')
    return img.addBands(ndvi);
}

var withNDVI1 = coll1.map(addNDVI);
var withNDVI2 = coll2.map(addNDVI);
var withNDVI3 = coll3.map(addNDVI);
var withNDVI4 = coll4.map(addNDVI);
var withNDVI5 = coll5.map(addNDVI);
var withNDVI6 = coll6.map(addNDVI);
var withNDVI7 = coll7.map(addNDVI);
var withNDVI8 = coll8.map(addNDVI);
var withNDVI9 = coll9.map(addNDVI);
var withNDVI10 = coll10.map(addNDVI);
var withNDVI11 = coll11.map(addNDVI);
var withNDVI12 = coll12.map(addNDVI);

var greenest1 =
withNDVI1.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest2 =
withNDVI2.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest3 =
withNDVI3.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest4 =
withNDVI4.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest5 =
withNDVI5.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest6 =
withNDVI6.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

```

```

var greenest7 =
withNDVI7.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).clip(dt);

var greenest8 =
withNDVI8.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).clip(dt);

var greenest9 =
withNDVI9.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).clip(dt);

var greenest10 =
withNDVI10.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).clip(dt);

var greenest11 =
withNDVI11.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).clip(dt);

var greenest12 =
withNDVI12.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).clip(dt);

var ndvi1 = greenest1.normalizedDifference(['B8','B4']).rename('NDVI1');
var ndvi2 = greenest2.normalizedDifference(['B8','B4']).rename('NDVI2');
var ndvi3 = greenest3.normalizedDifference(['B8','B4']).rename('NDVI3');
var ndvi4 = greenest4.normalizedDifference(['B8','B4']).rename('NDVI4');
var ndvi5 = greenest5.normalizedDifference(['B8','B4']).rename('NDVI5');
var ndvi6 = greenest6.normalizedDifference(['B8','B4']).rename('NDVI6');
var ndvi7 = greenest7.normalizedDifference(['B8','B4']).rename('NDVI7');
var ndvi8 = greenest8.normalizedDifference(['B8','B4']).rename('NDVI8');
var ndvi9 = greenest9.normalizedDifference(['B8','B4']).rename('NDVI9');
var ndvi10 = greenest10.normalizedDifference(['B8','B4']).rename('NDVI10');
var ndvi11 = greenest11.normalizedDifference(['B8','B4']).rename('NDVI11');
var ndvi12 = greenest12.normalizedDifference(['B8','B4']).rename('NDVI12');

var stacked_ndvi =
ndvi1.addBands([ndvi2,ndvi3,ndvi4,ndvi5,ndvi6,ndvi7,ndvi8,ndvi9,ndvi10,ndvi11,ndvi12]);

var stacked_ndvi_scaled = stacked_ndvi.multiply(100).add(100).uint8();

Map.addLayer(stacked_ndvi_scaled,{bands:['NDVI9','NDVI5','NDVI7'],min:100,max:200},'Stacked_NDVI');

Map.addLayer(greenest9,{bands:['B8','B4','B3'],min:0,max:5000},'Sen_Feb1FN');

print(ndvi1)

```

```

var gt = Crop.merge(other);
var training_data = stacked_ndvi_scaled.sampleRegions({
  collection : gt,
  properties:['class'],
  scale:10,
});
var classifier = ee.Classifier.smileRandomForest(100)
  .train({
    features:training_data,
    classProperty: 'class',
  });
var classified = stacked_ndvi_scaled.classify(classifier, 'Classified');
var classified1 = classified.updateMask(classified.eq(1))
Map.addLayer(classified1,{palette:['white','yellow'],min:0,max:1},'Classified
');
var areaImage = ee.Image.pixelArea().addBands(classified1);
var areas = areaImage.reduceRegion({
  reducer: ee.Reducer.sum().group({
    groupField: 1,
    groupName: 'class',
  }),
  geometry: dt,
  scale: 10,
  maxPixels: 1e13,
});
print('##### CLASS AREA SQ. METERS #####');
print(areas);
var trainAccuracy = classifier.confusionMatrix();
print('Resubstitution error matrix: ', trainAccuracy);
print('Training overall accuracy: ', trainAccuracy.accuracy());
Export.image.toDrive({
  image: classified1,

```

```
    description: 'Kharif_20221-22',  
    scale: 30,  
    folder: '2021-22',  
    region: dt  
  });  
Export.image.toDrive({  
  image: ndvil,  
  description: 'Kharif_20221-22NDVI',  
  scale: 30,  
  folder: '2021-22',  
  region: dt  
});
```

For Rabi crop:

```
function maskS2clouds(image) {
  var qa = image.select('QA60');
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
    .and(qa.bitwiseAnd(cirrusBitMask).eq(0));
  return image.updateMask(mask).divide(10000);
}

var dt = table.filter(ee.Filter.eq('DISTNAME', 'CHURU'));
var coll1 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-10-01', '2020-10-15')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
  .filterBounds(dt);
var coll2 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-10-16', '2020-10-31')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
  .filterBounds(dt);
var coll3 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-11-01', '2020-11-15')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
  .filterBounds(dt);
var coll4 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-11-16', '2020-11-30')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
  .filterBounds(dt);
var coll5 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
  .filterDate('2020-12-01', '2020-12-15')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .map(maskS2clouds);
  .filterBounds(dt);
```

```

var coll6 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
    .filterDate('2020-12-16','2020-12-31')
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
    .map(maskS2clouds);
    .filterBounds(dt);
var coll7 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
    .filterDate('2021-01-01','2021-01-15')
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
    .map(maskS2clouds);
    .filterBounds(dt);
var coll8 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
    .filterDate('2021-01-16','2021-01-31')
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
    .map(maskS2clouds);
    .filterBounds(dt);
var coll9 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
    .filterDate('2021-02-01','2021-02-15')
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
    .map(maskS2clouds);
    .filterBounds(dt);
var coll10 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
    .filterDate('2021-02-16','2021-02-28')
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
    .map(maskS2clouds);
    .filterBounds(dt);
var coll11 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
    .filterDate('2021-03-01','2021-03-15')
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
    .map(maskS2clouds);
    .filterBounds(dt);
var coll12 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
    .filterDate('2021-03-16','2021-03-31')

```

```

        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))

        .map(maskS2clouds);

        .filterBounds(dt);

    print(coll1);

var addNDVI = function(img){
    var ndvi = img.normalizedDifference(['B8','B4']).rename('NDVI')
    return img.addBands(ndvi);
}

var withNDVI1 = coll1.map(addNDVI);
var withNDVI2 = coll2.map(addNDVI);
var withNDVI3 = coll3.map(addNDVI);
var withNDVI4 = coll4.map(addNDVI);
var withNDVI5 = coll5.map(addNDVI);
var withNDVI6 = coll6.map(addNDVI);
var withNDVI7 = coll7.map(addNDVI);
var withNDVI8 = coll8.map(addNDVI);
var withNDVI9 = coll9.map(addNDVI);
var withNDVI10 = coll10.map(addNDVI);
var withNDVI11 = coll11.map(addNDVI);
var withNDVI12 = coll12.map(addNDVI);

var greenest1 =
withNDVI1.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest2 =
withNDVI2.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest3 =
withNDVI3.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest4 =
withNDVI4.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest5 =
withNDVI5.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

```

```

var greenest6 =
withNDVI6.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest7 =
withNDVI7.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest8 =
withNDVI8.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest9 =
withNDVI9.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cli
p(dt);

var greenest10 =
withNDVI10.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cl
ip(dt);

var greenest11 =
withNDVI11.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cl
ip(dt);

var greenest12 =
withNDVI12.qualityMosaic('NDVI').select(['B2','B3','B4','B8','B11','B12']).cl
ip(dt);

var ndvi1 = greenest1.normalizedDifference(['B8','B4']).rename('NDVI1');
var ndvi2 = greenest2.normalizedDifference(['B8','B4']).rename('NDVI2');
var ndvi3 = greenest3.normalizedDifference(['B8','B4']).rename('NDVI3');
var ndvi4 = greenest4.normalizedDifference(['B8','B4']).rename('NDVI4');
var ndvi5 = greenest5.normalizedDifference(['B8','B4']).rename('NDVI5');
var ndvi6 = greenest6.normalizedDifference(['B8','B4']).rename('NDVI6');
var ndvi7 = greenest7.normalizedDifference(['B8','B4']).rename('NDVI7');
var ndvi8 = greenest8.normalizedDifference(['B8','B4']).rename('NDVI8');
var ndvi9 = greenest9.normalizedDifference(['B8','B4']).rename('NDVI9');
var ndvi10 = greenest10.normalizedDifference(['B8','B4']).rename('NDVI10');
var ndvi11 = greenest11.normalizedDifference(['B8','B4']).rename('NDVI11');
var ndvi12 = greenest12.normalizedDifference(['B8','B4']).rename('NDVI12');

var stacked_ndvi =
ndvi1.addBands([ndvi2,ndvi3,ndvi4,ndvi5,ndvi6,ndvi7,ndvi8,ndvi9,ndvi10,ndvi11
,ndvi12]);

var stacked_ndvi_scaled = stacked_ndvi.multiply(100).add(100).uint8();

Map.addLayer(stacked_ndvi_scaled,{bands:['NDVI9','NDVI5','NDVI3'],min:100,
max:200},'Stacked_NDVI');

```

```

Map.addLayer(greenest9,{bands:['B8','B4','B3'],min:0,max:5000},'Sen_Feb1FN');
print(ndvi1)
var gt = cropr_2021.merge(otherr_2021);
var training_data = stacked_ndvi_scaled.sampleRegions({
  collection : gt,
  properties:['class'],
  scale:10,
});
var classifier = ee.Classifier.smileRandomForest(100)
  .train({
    features:training_data,
    classProperty: 'class',
  });
var classified = stacked_ndvi_scaled.classify(classifier, 'Classified');
var classified1 = classified.updateMask(classified.eq(1))
Map.addLayer(classified1,{palette:['white','yellow'],min:0,max:1},'Classified
');
var areaImage = ee.Image.pixelArea().addBands(classified1);
var areas = areaImage.reduceRegion({
  reducer: ee.Reducer.sum().group({
    groupField: 1,
    groupName: 'class',
  }),
  geometry: dt,
  scale: 10,
  maxPixels: 1e13,
  //tileScale: 8
});
print('##### CLASS AREA SQ. METERS #####');
print(areas);
var trainAccuracy = classifier.confusionMatrix();
print('Resubstitution error matrix: ', trainAccuracy);

```

```
print('Training overall accuracy: ', trainAccuracy.accuracy());  
Export.image.toDrive({  
    image: classified1,  
    description: 'Rabi_2020-21',  
    scale: 30,  
    folder:'2020-21a',  
    region: dt  
});  
Export.image.toDrive({  
    image: ndvil1,  
    description: 'Rabi_2020-21NDVI',  
    scale: 30,  
    folder:'2020-21a',  
    region: dt  
});
```