End Sem Project Report

Richard Joe - 2110110425
Aryan Tiwari - 2110110140
Siddharth Nadupalli - 2110110500

Table of Contents

Project Structure

Overview

The structure and folders have been maintained on - Google Drive
The **Code FIles** are run on **Google Colab** itself for easier integration after mounting
Orthomosaic Files: 10
Training CSV: 11 (1 for each orthomosaic and 1 combined)
Prediction CSV: 10
Training Kmls created: 10
Prediction Kmls created: 10

Code_Files	me	Oct 1, 2024 me	—
csv_backups	me	Oct 1, 2024 me	—
Grid_Images	me	Oct 2, 2024 me	—
Model_weights	me	Oct 1, 2024 me	—
Notes	me	Nov 5, 2024 me	—
Output_Files	me	Oct 22, 2024 me	—
sub-1-6	me	Jan 1, 1980 me	—
sub-8-13	me	Jan 1, 1980 me	—
sub-15-18	me	Jan 1, 1980 me	—
sub-19-21	me	Jan 1, 1980 me	—

1. **Code_Files**: Contains the .ipynb files with the code. Each file handles a different task.
2. **csv_backups**: Contains the google sheets and csv files, which may be the training data, the path for predictions, testing data etc
3. **Grid_Images**: Contains the orthomosaic files after overlaying the grid lines
4. **Model_weights**: Weights for pre-trained models of VGG16 and Inception_V3
5. **Notes**: Documentation to comprehend the code structure and project
6. **Orthomosaic**: All 10 orthomosaic files for the grids.
7. **Output_Files**: All 21 Kmls as highlighted before
8. **Sub-1-6**: Cropped Images from the Orthomosaic Grid_1_to_6. The images are cropped with a height of 352 and width of 261.
    a. Similarly other Subgrids are also done.

Code_Files

625__TIF_File.ipynb	me	Oct 1, 2024 me	3.6 MB
Crop_Grid.ipynb	me	Nov 24, 2024 me	474 KB
CSV_Creation.ipynb	me	5:32 PM me	2 KB
final_crop_area_estimation.ipynb	me	Nov 19, 2024 me	1,004 KB
Tiff_Extraction.ipynb	me	Nov 12, 2024 me	6 KB

1. **Crop_Grid.ipynb**: Creates the white grid overlay on orthomosaic images and then crops the grids according to the created overlay.
2. **CSV_Creation.ipynb**: Creates CSV File of all the cropped images in a folder (sub-1-6). 2 Types:
    a. For training: 2 Folders will be there within Fallow and LandUse. This will be reflected in the label column of csv: 1 for Fallow and 0 for LandUse.
    b. For prediction: All paths of the cropped images are generated and added to the path column in the csv.
3. **final_crop_area_estimation.ipynb**: Acts as the major chunk of the project which contains, Train-test split, preprocessing, model training, predictions, plotting, visual representation.
4. **KML_generation.ipynb**: Creates the kmls for training as well as prediction data using the training and csv files generated.
5. **Tiff_Extraction.ipynb**: Additional code to analyze the corner coordinates of the Tiff File / Orthomosaic. To be used later in sync with QGIS.

Data

The dataset is 10 Orthomosaic images of the village plots. Each file above 500MB in size contains at least 2 or 3 Grids adjoined together.

GRID_1_to_6.tif	me	Oct 18, 2024 me	565.3 MB	⋮
GRID_8_to_13.tif	me	Oct 18, 2024 me	1.04 GB	⋮
Grid_15_to_18.tif	me	Oct 18, 2024 me	507 MB	⋮
Grid_19_to_21.tif	me	Oct 18, 2024 me	345 MB	⋮
GRID_24_to_33.tif	me	Oct 18, 2024 me	1.96 GB	⋮
Grid_35_to_42.tif	me	Oct 18, 2024 me	1,013 MB	⋮
Grid_45_to_50_and_53.tif	me	Oct 18, 2024 me	839.1 MB	⋮
GRID_51_and_54_to_58.tif	me	Oct 18, 2024 me	1.73 GB	⋮
GRID_61_to_72.tif	me	Oct 18, 2024 me	1.95 GB	⋮
GRID_73_to_78.tif	me	Oct 18, 2024 me	1.91 GB	⋮

Training Data

Training CSV - 11

1. Training CSV for each grid - 10

path	label	crop
/content/drive/My Drive/sub-1-6/subimage_16_37.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_13_59.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_18_18.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_12_3.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_8_12.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_13_17.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_21_34.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_1_67.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_3_15.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_4_50.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_12_1.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_10_23.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_11_54.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_14_68.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_9_39.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_12_83.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_2_79.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_5_28.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_12_13.png	0	LandUse
/content/drive/My Drive/sub-1-6/subimage_2_55.png	0	LandUse

2. Compiled training CSV - 1

84	/content/drive/My Drive/Wassan/subimages/sub-19-21/subimage_12_52.png	0	not_tomato
85	/content/drive/My Drive/Wassan/subimages/sub-19-21/subimage_12_53.png	0	not_tomato
86	/content/drive/My Drive/Wassan/subimages/sub-19-21/subimage_12_54.png	0	not_tomato
87	/content/drive/My Drive/Wassan/subimages/sub-19-21/subimage_12_55.png	0	not_tomato
88	/content/drive/My Drive/Wassan/subimages/sub-19-21/subimage_12_56.png	0	not_tomato
89	/content/drive/My Drive/Wassan/subimages/sub-19-21/subimage_12_57.png	0	not_tomato
90	/content/drive/My Drive/Wassan/subimages/sub-19-21/subimage_12_58.png	0	not_tomato
91	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_4_30.png	1	tomato
92	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_5_31.png	1	tomato
93	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_7_31.png	1	tomato
94	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_5_32.png	1	tomato
95	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_7_32.png	1	tomato
96	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_14_23.png	1	tomato
97	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_15_23.png	1	tomato
98	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_2_69.png	1	tomato
99	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_2_70.png	1	tomato
100	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_20_78.png	1	tomato
101	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_18_77.png	1	tomato
102	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_14_17.png	0	not_tomato
103	/content/drive/My Drive/Wassan/subimages/sub-15-18/subimage_18_37.png	0	not_tomato

Training KML - 10


```
<?xml version="1.0" encoding="utf-8" ?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document id="root_doc">
<Schema name="m33" id="m33">
    <SimpleField name="Class" type="int"></SimpleField>
</Schema>
<Folder><name>m33</name>

  <Placemark>
    <Style>
      <LineStyle><color>ff0000ff</color></LineStyle>
      <PolyStyle><fill>0</fill></PolyStyle>
    </Style>
    <ExtendedData>
      <SchemaData schemaUrl="#m33">
        <SimpleData name="Class">1</SimpleData>
      </SchemaData>
    </ExtendedData>
    <MultiGeometry>
      <Polygon>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              78.22861297317534,13.962597700334207 78.22876687317535,13.962597700334207 78.22876687317535,13.962396240334208 78.22861297317534,13.962396240334208
78.22861297317534,13.962597700334207
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </MultiGeometry>
  </Placemark>
```


Data Processing

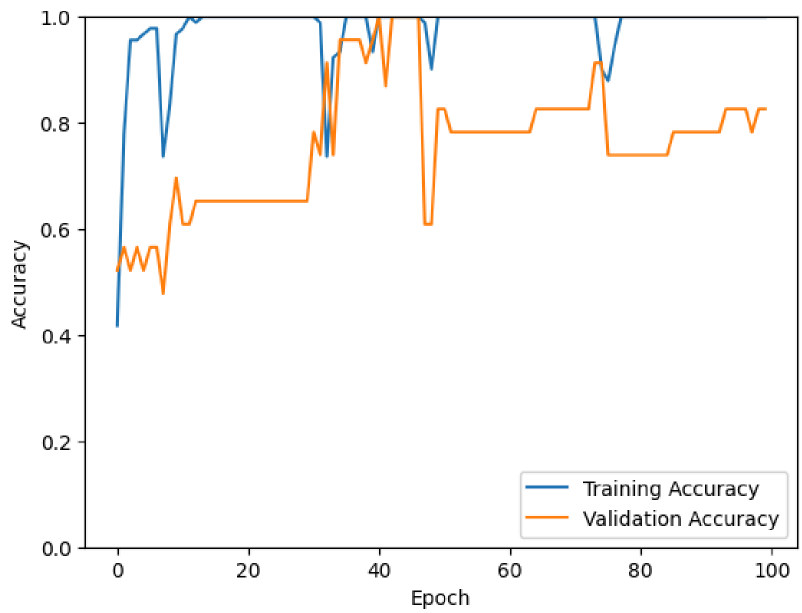
1. The tif images are cropped into subimages and stored in respective folders like sub-1-6 etc.
2. Each Subimage folder is crawled to generate a csv file of the paths of all images.
3. Training data is generated by cross checking the paths, the subimages and manually moving it into a training csv for each orthomosaic.
4. A training KML is generated from each csv for visualization purposes in QGIS software.
5. A compiled training csv is also generated to be used as the primary source of training data when predictions are being run on the grids.

Exploratory Analysis

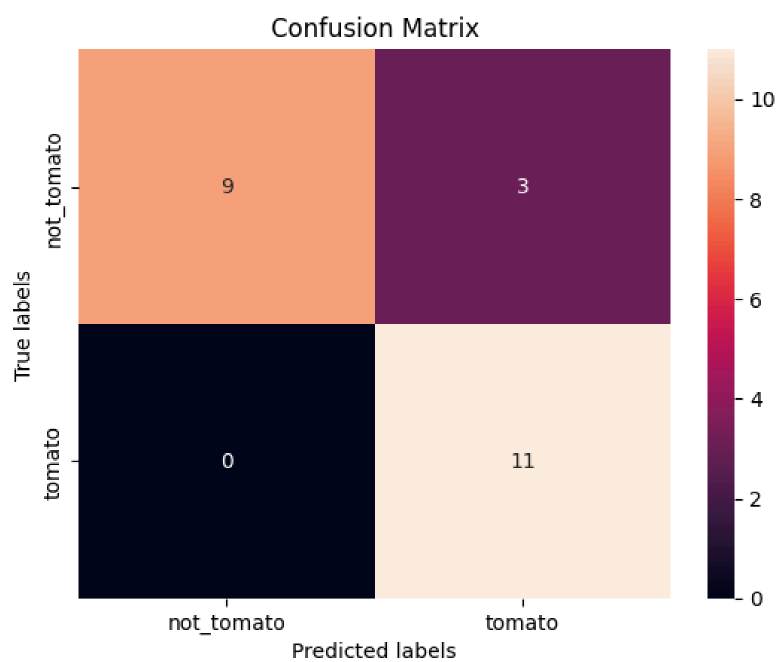
Models Explored

CD-CNN

1. Crop Estimation

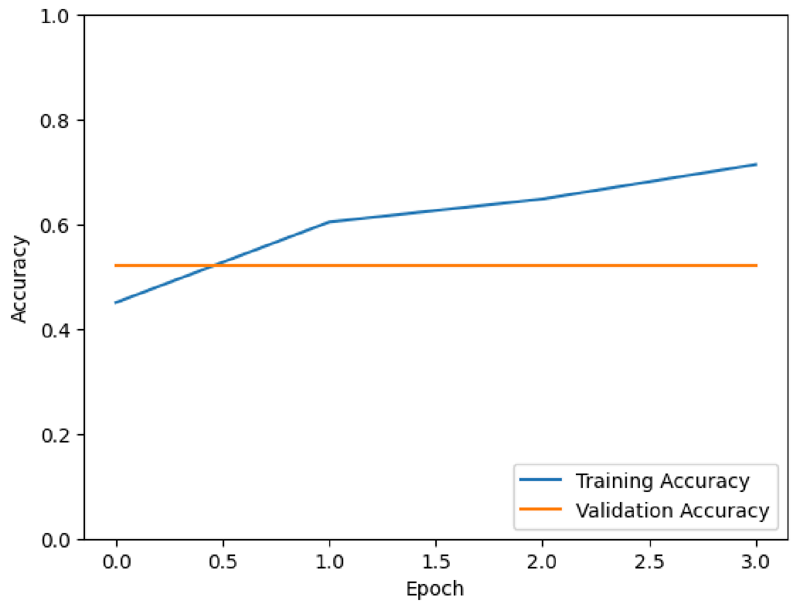


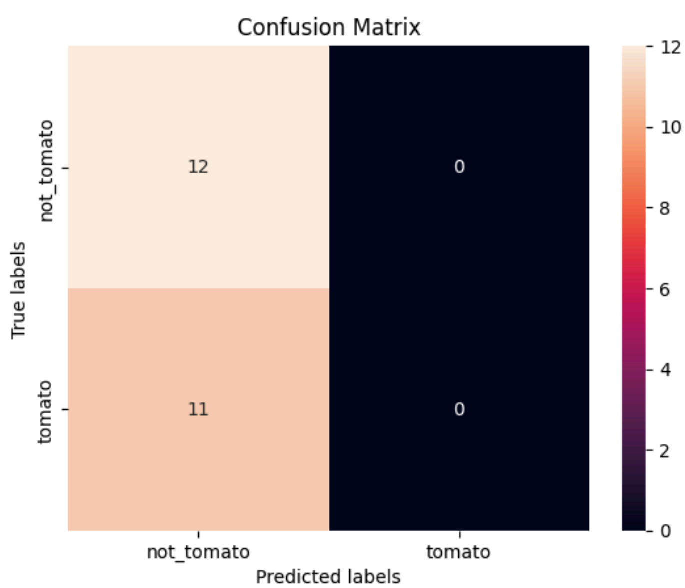
Accuracies VS Epochs for CD-CNN



Custom Model

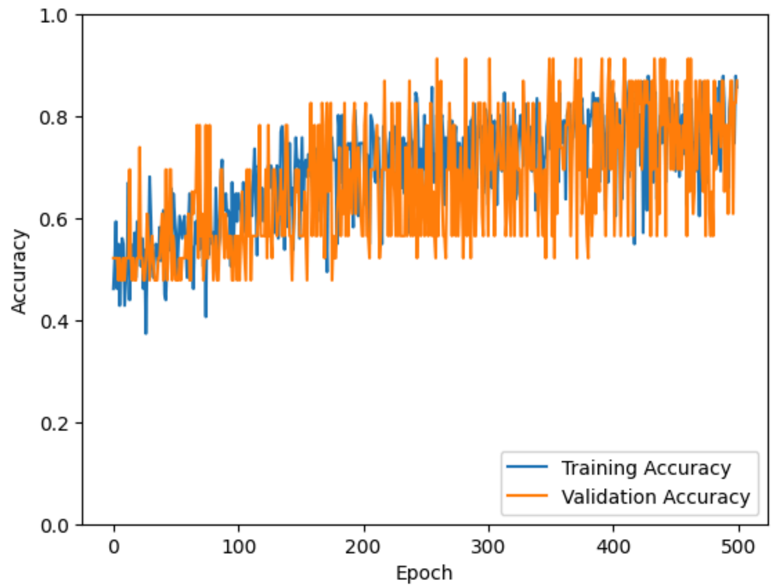
1. Crop Estimation

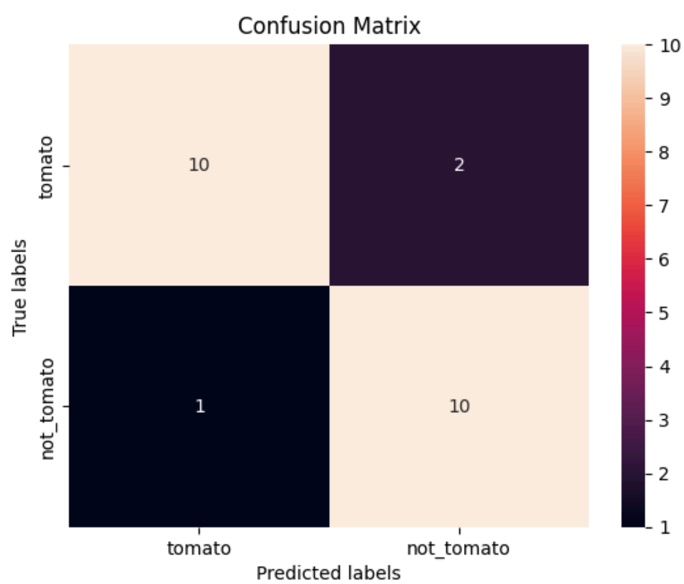




Resnet50

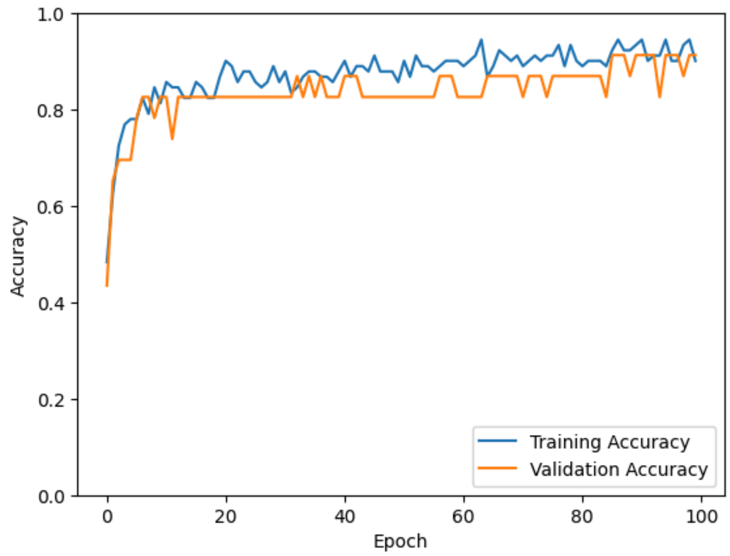
1. Crop Estimation

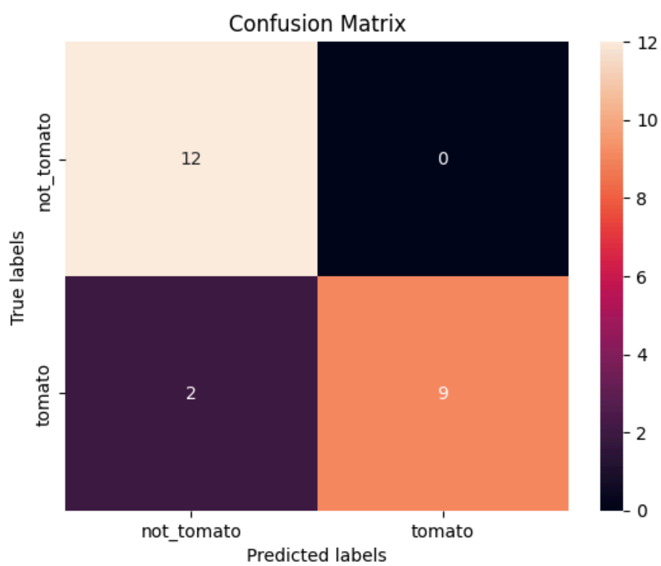




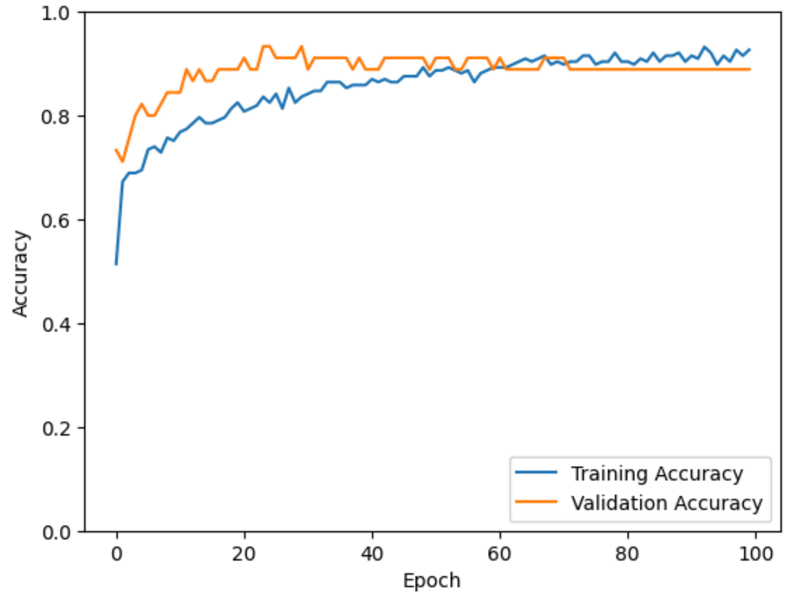
VGG16

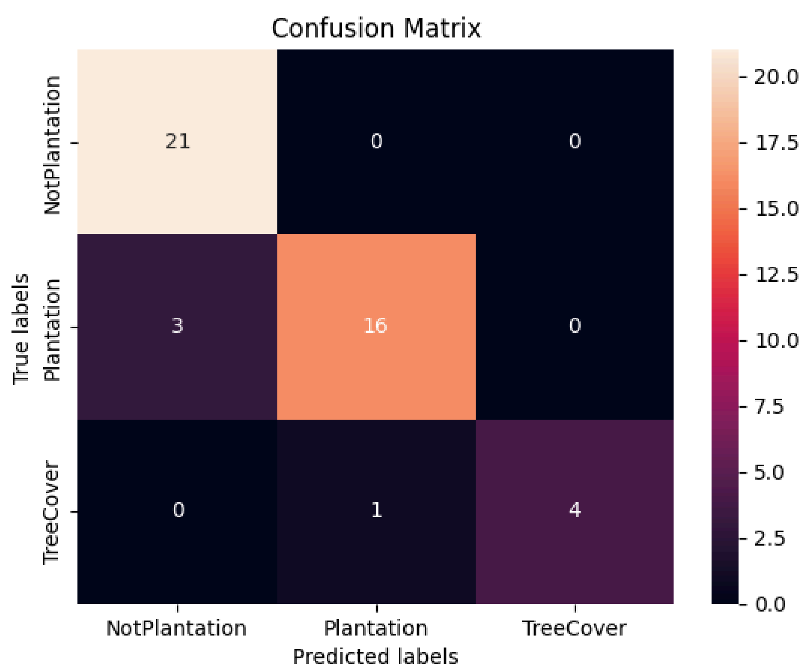
1. Crop Estimation





2. Plantation





YOLO

1. Plantation
    - YOLOv8N

      <u>Training</u>

```
Validating runs/detect/train5/weights/best.pt...
Ultralytics 8.3.21 🚀 Python-3.10.12 torch-2.5.0+cu121 CPU (Intel Xeon 2.20GHz)
Model summary (fused): 168 layers, 3,006,233 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:05<00:00,  5.20s/it]
                   all         20         55    0.00636      0.446      0.147     0.0658
                  dead          2          2          0          0          0          0
                  good         17         41     0.0156      0.756      0.405      0.183
               stunted          7         12     0.0035      0.583     0.0362     0.0141
Speed: 1.7ms preprocess, 216.1ms inference, 0.0ms loss, 18.5ms postprocess per image
Results saved to runs/detect/train5
```

      <u>Validation</u>

```
[ ] model.val(save_json=True)

Ultralytics 8.3.21 🚀 Python-3.10.12 torch-2.5.0+cu121 CPU (Intel Xeon 2.20GHz)
Model summary (fused): 168 layers, 3,006,233 parameters, 0 gradients, 8.1 GFLOPs
val: Scanning /content/drive/MyDrive/data-code/yolo_1-6_train-label_29/labels/val.cache... 20 images,
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|
                   all         20         55    0.00636      0.446      0.147     0.0658
                  dead          2          2          0          0          0          0
                  good         17         41     0.0156      0.756      0.405      0.183
               stunted          7         12     0.0035      0.583     0.0362     0.0141
Speed: 1.8ms preprocess, 230.2ms inference, 0.0ms loss, 19.2ms postprocess per image
Saving runs/detect/train52/predictions.json...
```

    - YOLOv8S

      <u>Training</u>

```
Ultralytics 8.3.27 🚀 Python-3.10.12 torch-2.5.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11,126,745 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  2.70it/s]
                   all         25         70      0.485       0.14      0.117     0.0499
                  dead          3          3          1          0          0          0
                  good         22         49      0.299      0.365      0.314      0.137
               stunted         11         18      0.155     0.0556     0.0389     0.0129
Speed: 0.3ms preprocess, 3.4ms inference, 0.0ms loss, 2.9ms postprocess per image
Results saved to /content/drive/MyDrive/yolo_runs/exp12
```

      <u>Validation</u>

```
model.val()

Ultralytics 8.3.27 🚀 Python-3.10.12 torch-2.5.0+cu121 CPU (Intel Xeon 2.20GHz)
Model summary (fused): 168 layers, 11,126,745 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/drive/MyDrive/data-code/yolo_1-6_train-label_29/labels/val.cache... 21 images, 0 backgrounds, 0 co
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:13<00:
                   all         21         58      0.776      0.217      0.127      0.065
                  dead          2          2          1          0          0          0
                  good         18         43      0.327      0.651      0.363      0.187
               stunted          8         13          1          0     0.0173     0.0078
Speed: 1.8ms preprocess, 592.1ms inference, 0.0ms loss, 35.0ms postprocess per image
Results saved to runs/detect/train2
ultralytics.utils.metrics.DetMetrics object with attributes:
```

Methodology

1. Gridding & Cropping

1. Splits the orthomosaic tif file into even grids of height of 352 and width 261.
    a. Initially displays a white grid overlay indicative of how the large tif file would be cropped into separate subimages. (subimages are a .png file)
    b. The number of rows and columns for gridding must be updated by trial and error for each orthomosaic depending on dimensions of the tiff image.
2. After Gridding is generated in the required format the orthomosaic is then cropped accordingly.
    a. The cropped .png files are stored in separate folders for each orthomosaic
    Eg: sub-1-6 - Folder containing cropped subimages of Grid1-6 Orthomosaic image.

Input:

1. Orthomosaic Image / Tif File

Output:

1. White Grid Overlay on Orthomosaic (after Gridding)
2. Sub Folders with cropped images (after Cropping)

2. CSV Generation

1. The created subfolder is crawled by the script and a csv file is generated containing the paths to the subimages.
    a. The paths are stored under the column name “path”.

Eg:

path
/content/drive/My Drive/sub-1-6/subimage_6_55.png

Input:

1. Subimage Folders

Output:

1. CSV file containing paths

3. Training the Models

1. From the CSV file for each Subimage folder, around 20 paths are moved into another Subimage Training CSV file
    a. Each corresponding Subimage is also analyzed and labelled accordingly.

Eg:

path	label	crop
/content/drive/My Drive/sub-1-6/subimage_16_37.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_13_59.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_18_18.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_12_3.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_8_12.png	1	Fallow
/content/drive/My Drive/sub-1-6/subimage_13_17.png	0	LandUse

2. All the individual training CSV files are combined to form a Compiled Training CSV.
    a. The Compiled Training CSV is then used for training the model. This expansion in training data is necessary to prevent underfitting.
3. The Compiled Training CSV is then used to train the different models
    a. Some models require pretrained weights as well.
        i. Inception V3
           Eg:

inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5	me	Oct 1, 2024 me	83.8 MB

        ii. VGG16
           Eg:

vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5	me	Oct 1, 2024 me	56.2 MB

Input:

1. Subimage Folder
2. Pretrained Model Weights

Output:

1. Training CSV File for each SubImage Folder
2. Compiled Training CSV File

4. Predictions on Orthomosaic images

1. The Subimages picked up based on the paths of the CSV files are converted to np arrays for processing
    a. They are converted to an array with 3 dimensions.
    b. They are resized to 32x32 for every model and 75x75 for Inception V3.
    c. The 3rd dimension indicates the 3 channels of RGB
    d. They are then normalized by a factor of 255.
2. The models predict a label for each subimage, may it be:
    a. Tomato vs Not tomato
    b. Plantation vs Tree Cover vs Not plantation
    c. Fallow vs LandUse
3. The predictions are stored in a details array along with their row number and column number
    a. The row number and column number is extracted from the path of the Subimage using delimiters and appropriate splits.
4. This details array is then used to generate a colored grid overlay for that particular orthomosaic grid.
    a. Tomato - Red vs Not tomato - white
    b. Plantation - Red vs Tree Cover - Green vs Not Plantation - white
    c. Fallow - Red vs LandUse - white

Input:

1. CSV File for Orthomosaic SubFolder

Output:

1. Predicted Grid Overlay

KML Generation

1. Identifying the top-left latitude and longitude of an Orthomosaic image. 2 approaches were used:
    a. The Tif file was uploaded on QGIS software to obtain to top-left coordinates
    b. The Tiff_Extraction.ipynb script was used to obtain the top-left coordinates using the library Gdal.
2. The Cell width and Cell height are converted into latitude and longitude measures
    a. The coordinate points of top-left and bottom-right points of the orthomosaic are identified and then the number of rows and columns are used to arrive at the solution.
3. The KMLs are generated with a placemark for each subgrid.
    a. Classes are indicated within each placemark for each subgrid
    b. The coordinates are indicated as a Polygon -> Linear Ring.

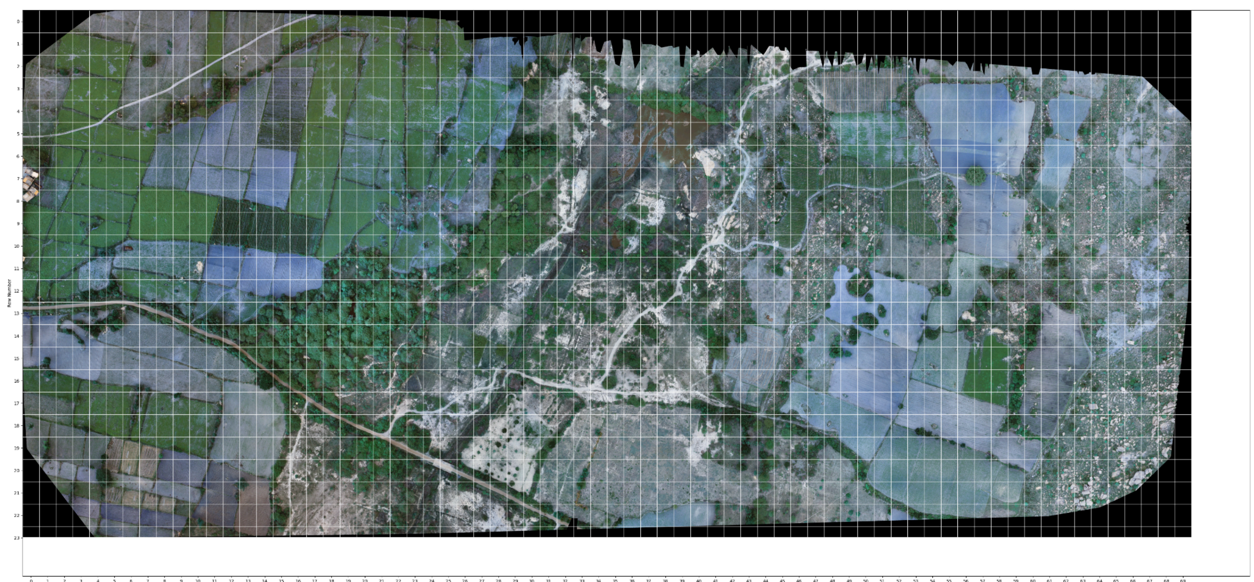
Visualization in QGIS

1. All the 10 Orthomosaic images are added to QGIS, and the training and prediction KMLs, with coloured grids are added as well, to overlay the tiff images. This will give a proper visualization of the whole village, with predicted areas of a specific crop, along with the training dataset as well.

Output

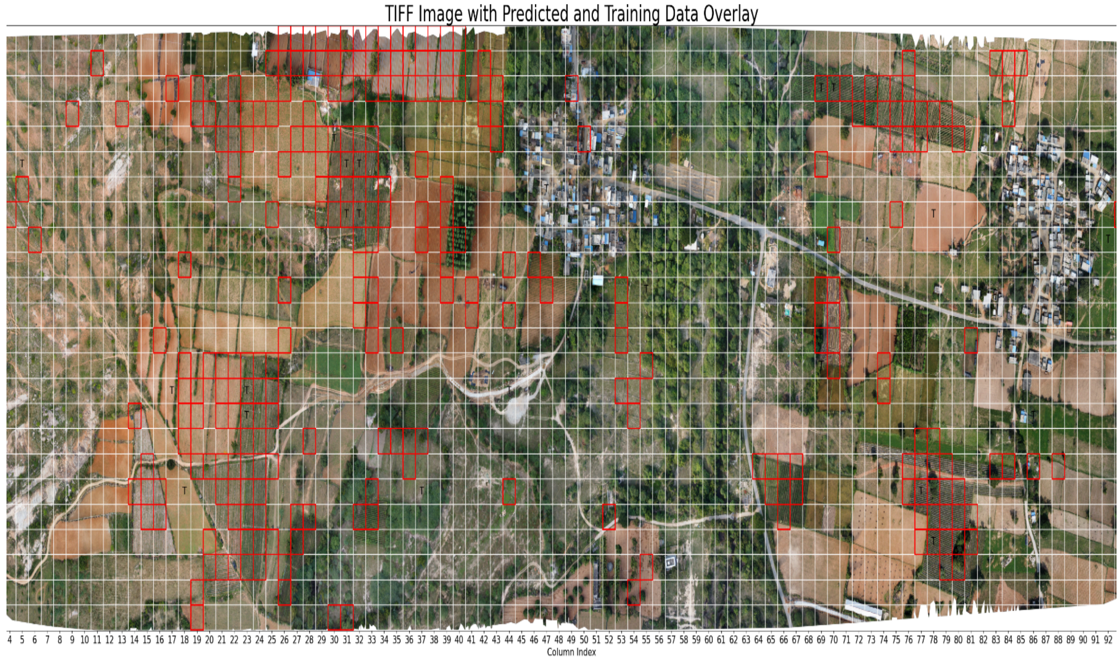
Visualization

White Grid Overlay - Subimages

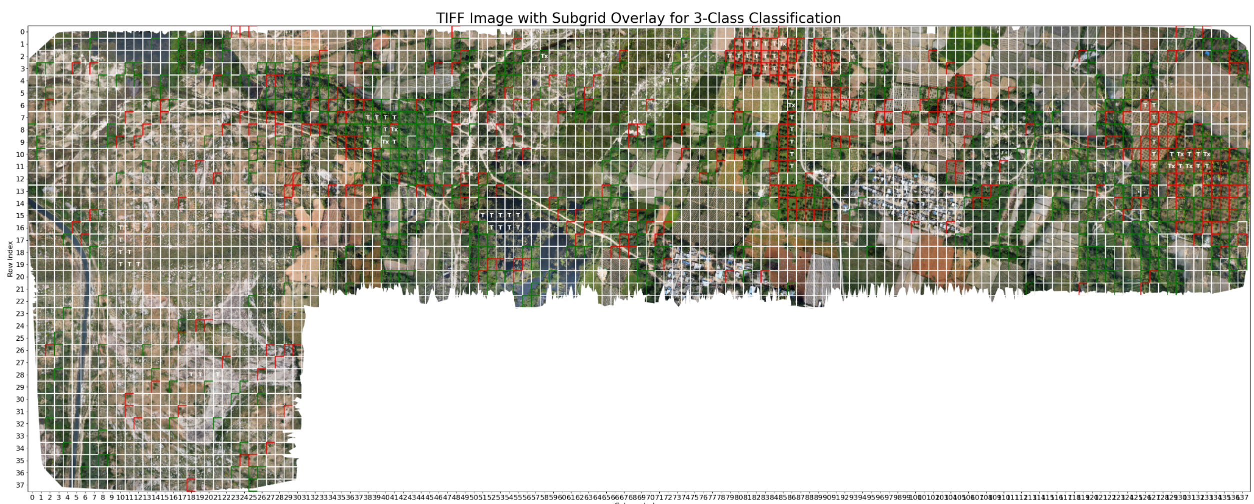


Colored Grid Overlay - Prediction

1. Crop Estimation



2. Plantation



YOLO - Identification

1. Plantation

Files

Predicted KML Files


```
<?xml version="1.0" encoding="utf-8" ?>
    <kml xmlns="http://www.opengis.net/kml/2.2">
    <Document id="root_doc">
    <Schema name="predictions" id="predictions">
        <SimpleField name="Class" type="int"></SimpleField>
    </Schema>
    <Folder><name>Predicted Grid</name>

      <Placemark>
        <Style>
          <LineStyle><color>ffffffff</color></LineStyle>
          <PolyStyle><fill>0</fill></PolyStyle>
        </Style>
        <ExtendedData>
          <SchemaData schemaUrl="#predictions">
            <SimpleData name="Class">0</SimpleData>
          </SchemaData>
        </ExtendedData>
        <MultiGeometry>
          <Polygon>
            <outerBoundaryIs>
              <LinearRing>
                <coordinates>
                  78.22676617317535,13.96453669104029 78.22692480176583,13.96453669104029 78.22692480176583,13.964328088117936
78.22676617317535,13.964328088117936 78.22676617317535,13.96453669104029
                </coordinates>
              </LinearRing>
            </outerBoundaryIs>
          </Polygon>
        </MultiGeometry>
      </Placemark>
```


Societal Impact

This project addresses the critical challenge of estimating agricultural yield and land usage, offering a more efficient and accurate alternative to traditional methods. Its implications extend across multiple dimensions, fostering sustainability, policy improvement, and modern agricultural practices.

1. Sustainable Initiatives and Environmental Responsibility

- **Land Management:**
  The system identifies underutilized land and fallow areas, aiding in optimal land management and maximizing agricultural productivity.
- **Ecological Balance:**
  By monitoring land use effectively, the project discourages over-exploitation of resources and promotes agroforestry and other sustainable practices.

2. Policy, Planning, and Economic Transparency

- **Policy Formulation:**
  Accurate yield estimates enable governments to create better-informed policies for agricultural subsidies, disaster relief, and food security planning.
- **Fair Pricing:**
  Transparent data on crop yield facilitates equitable crop pricing, ensuring fair compensation for farmers and reducing conflicts in the agricultural sector.

3. Data-Driven and Efficient Farming

- **Automation and Efficiency:**
  Automating yield estimation reduces the need for labor-intensive manual surveys, significantly saving time and resources while minimizing errors.
- **Empowering Farmers:**
  Visualized insights empower farmers to make informed decisions on crop rotation, irrigation, and planting strategies, enhancing efficiency and sustainability.

4. Precision Agriculture and Technological Integration

- **Targeted Resource Usage:**
  Integrating with IoT and real-time sensors allows for precise application of fertilizers, pesticides, and water, reducing waste and environmental harm.
- **Modern Farming Practices:**
  The system supports the adoption of advanced farming technologies, bringing the benefits of AI-driven insights to remote and underdeveloped agricultural areas.

Future Scope

The project demonstrates significant potential for further development and enhancement to make it more robust, user-friendly, and widely applicable. Key areas for future work include:

1. **Enhancing Model Accuracy:**

   - Increasing the size and diversity of the training dataset to improve the generalization and predictive accuracy of the models.
   - Refining the VGG16 model and experimenting with other advanced models such as ResNet80 to explore their performance in predicting agricultural parameters.
2. **Improving Geospatial Precision:**
   - Generating KML files with highly precise coordinates to ensure better grid plotting and more accurate visualization.
   - Optimizing the integration of KML files with QGIS for seamless geospatial analysis.
3. **Developing User-Friendly Applications:**
   - Creating a web-based platform or a user-interactive mobile application that allows users to upload orthomosaic images and receive detailed visualizations and predictions as output.
   - Incorporating features for real-time processing and result sharing to facilitate practical use in field surveys and decision-making.
4. **Expanding Use Cases:**
   - Extending the application to other domains, such as forestry for monitoring tree cover, or urban planning for land use assessment.
   - Integrating multi-spectral or hyper-spectral imaging data to analyze additional parameters, such as soil health or water stress.
5. **Increasing Scalability:**
   - Deploying the system on cloud-based platforms to handle large datasets and support concurrent users.
   - Collaborating with agricultural organizations to scale the solution to larger regions and diverse environmental conditions.