NAAMII

Food and Agriculture Organization
of the United Nations

TEAMFARM

TRANSFORM AGRICULTURE WITH AI

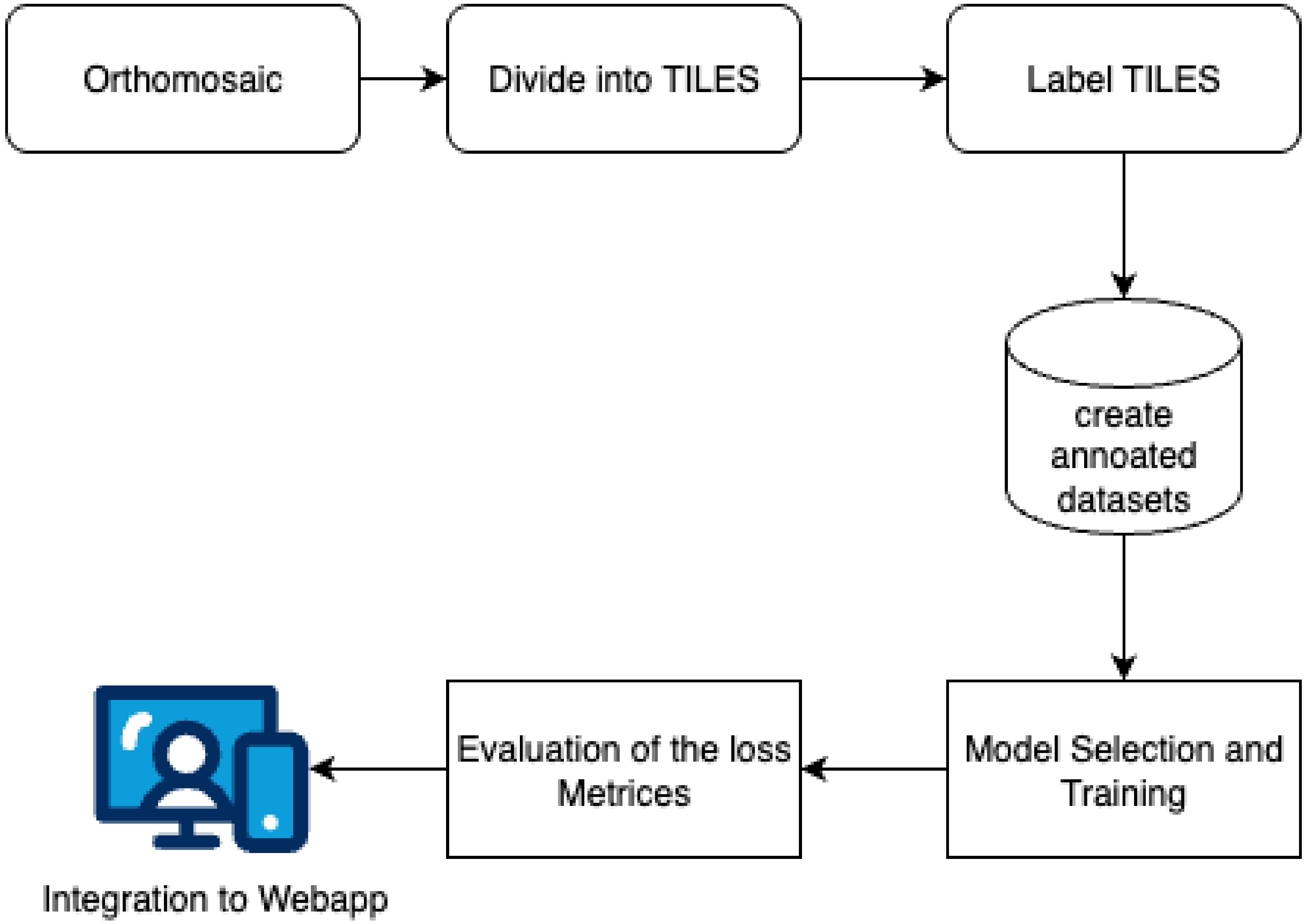
Transforming Agriculture with Technology

Problem Statement

To develop an AI model to properly segment between a farmland and non-farm land

To do the damage assesment of the farmland

IDEATION



Orthomosaic
Divide into TILES
Label TILES
create annotated datasets
Model Selection and Training
Evaluation of the loss Metrices
Integration to Webapp

Stage 1 Orthomosiac and Tiling

Orthomosiac

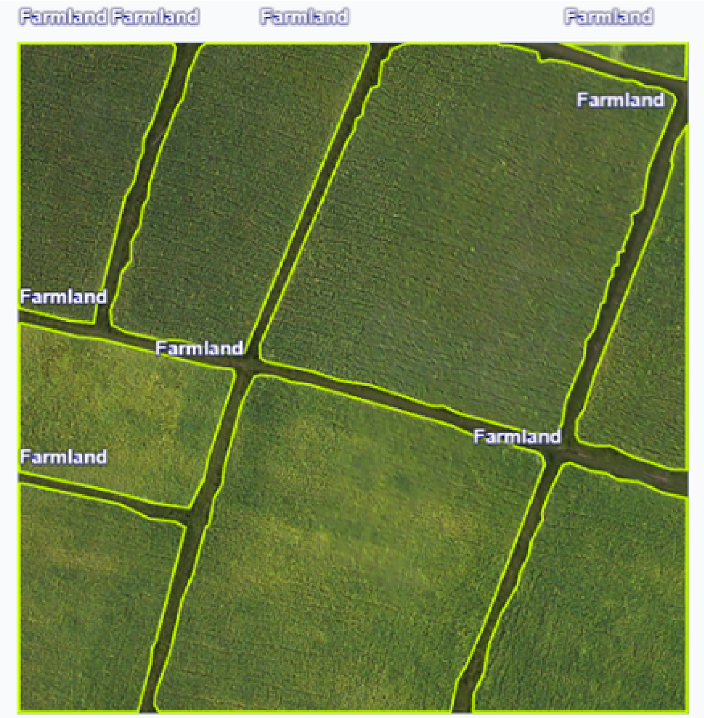
Tiling into 1024 * 1024 pixels

Sample Tiles after cleaning

Stage 2 Annotation

- Annotation is done using roboflow
- Classes:
  - Farmland
  - Vegetation(Barrenland, Bush, Tree)
  - Sand

annotate



Pros

- SAM & Label Assist speed up labeling.
- Easy accessible for training platforms with scripts
- Great for collaboration and dataset versioning.

Cons.

- Requires stable internet (cloud dependent).
- Makes datasets public

Stage 3 Model Selection and Training

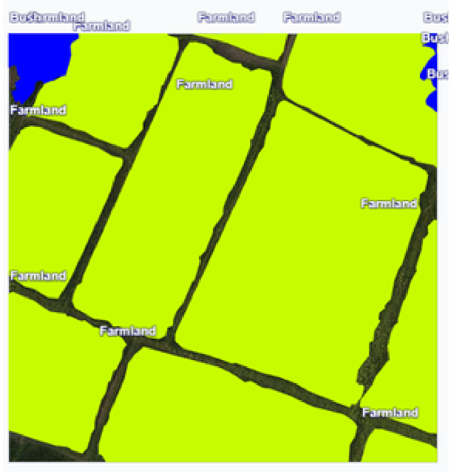
1. Understanding the Problem & Data
   - Clearly defined the task.
   - Analyzed dataset quality, class distribution, and preprocessing needs.

2. Literature & Model Review
   - Reviewed existing models used in similar agricultural or segmentation tasks.
   - Compared strengths/limitations of classical and deep learning approaches.

3. Initial Experiments with Simple Models
   - Trained on ~50 annotated images to get a quick baseline.
   - Observed issues like poor generalization and inconsistent masks.

4. Iterative Model Switching & Testing
   - Experimented with multiple architectures (e.g., U-Net, SegFormer).
   - Evaluated each model's accuracy, inference speed, and robustness on field conditions.
   - Identified limitations such as small dataset sensitivity, lighting variations, and misclassification.

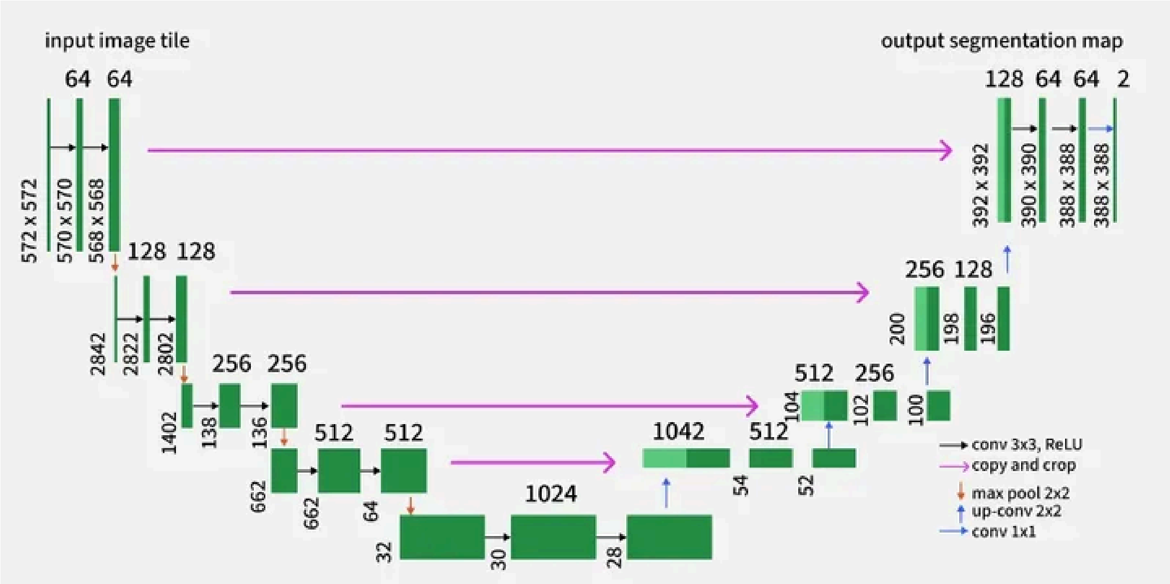
5 Selecting the Best Model
   - Based on experimental results and error analysis, chosed the model that showed:
   - Highest stability across conditions
   - Best performance on validation samples
   - Most practical to deploy for the project

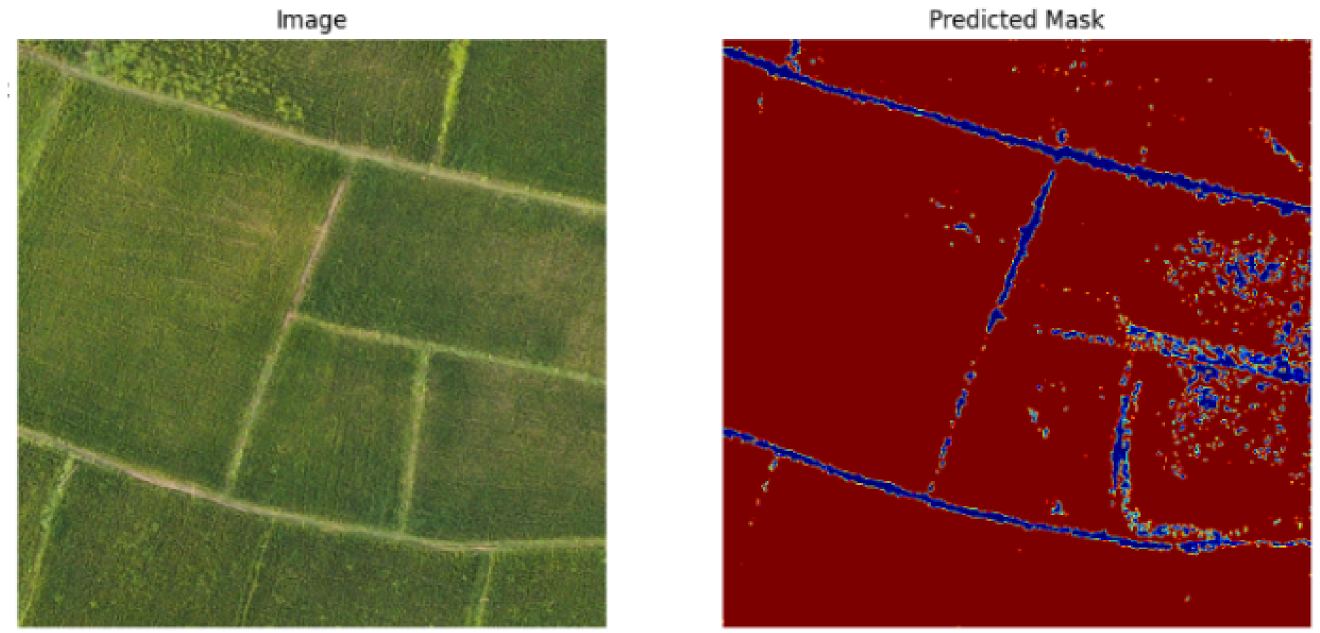


progress till now

Experiment with Different Models

- ## U-Net
  - Since UNet is a widely used and reliable architecture for image segmentation, we chose it as our initial baseline model.
  - Strong performance in many biomedical and remote-sensing tasks





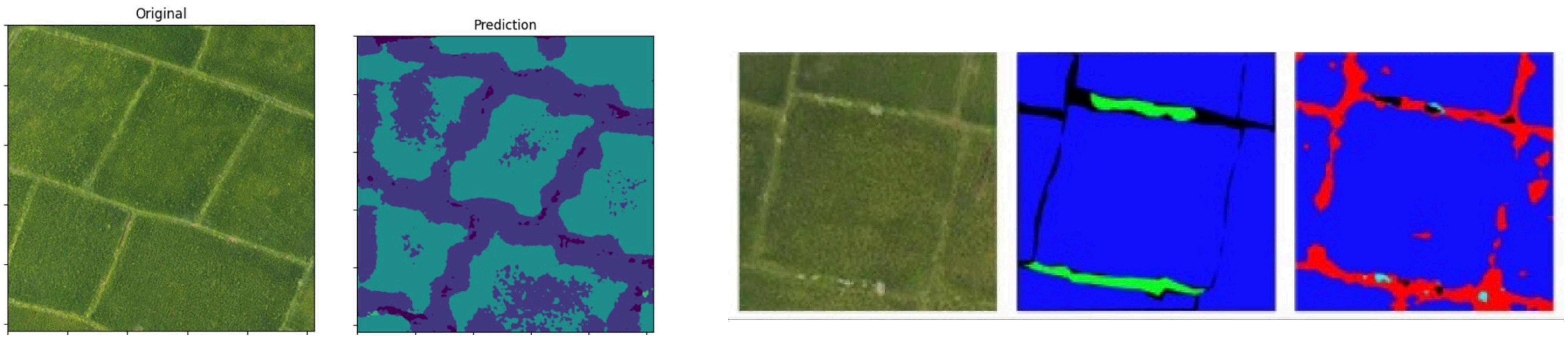
Model Setup

- Architecture: UNet
- Number of Classes: 7
- Dataset Size: 69 images
  - Training Set: 55 images
  - Validation Set: 14 images

Training Configuration

- Total Epochs: 50
- Reported Epoch: 7
- Metrics at Epoch 7:
- Training Loss: 1.3156
  - Validation Loss: 1.5828
  - Dice Score: 0.1121

- **Segformer**
  - Strong global feature extraction using Transformers
  - Lightweight decoders for fast inference
  - Robust performance on remote-sensing and outdoor scene datasets



Sample results from Segformer

- **U-NET with VGG16 encoder**
  - Encoder backbone (VGG16) provides strong feature extraction.
  - Better at capturing texture and spatial patterns than plain UNet.

Model Setup

- Architecture: UNet with VGG16 encoder
- Number of Classes: 5
  - Background
  - Barrenland
  - Vegetation
  - Farmland
  - Sand

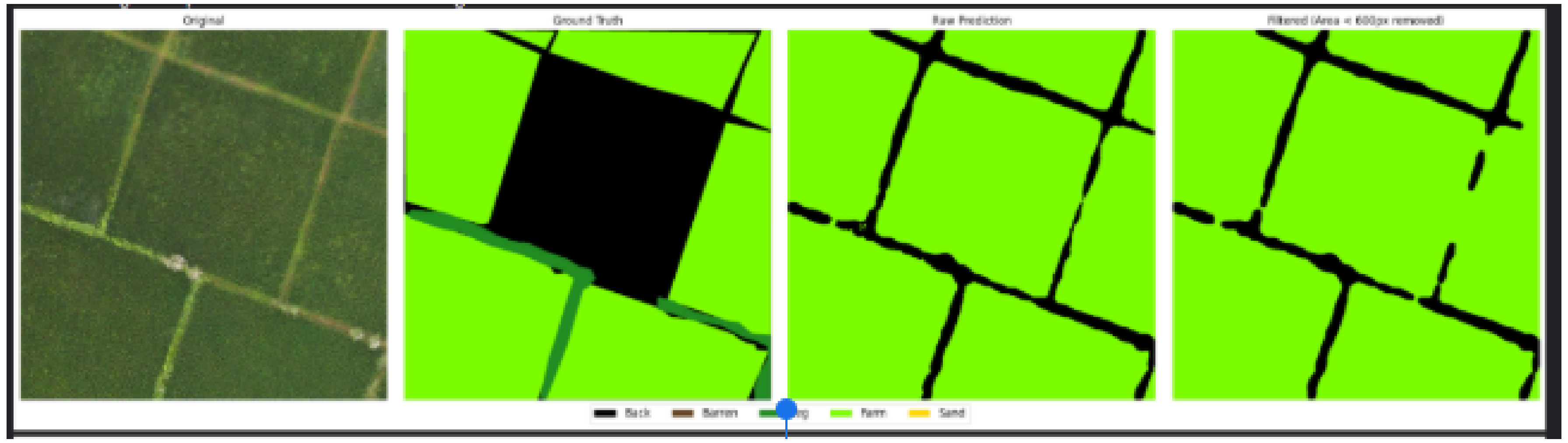
Dataset

- Total Images: 516
  - Train: 443
  - Validation: 48
  - Test: 25

Class	IoU	Dice (F1)	Precision	Recall
Background	0.4298	0.6012	0.5369	0.6831
Barrenland	0.753	0.8591	0.9264	0.8009
Vegetation	0.4014	0.5729	0.8061	0.4444
Farmland	**0.8898**	**0.9417**	0.901	0.9863
Sand	0.7105	0.8307	0.851	0.8114

Overall Metrics

- Mean IoU (mIoU): 0.6369
- Mean Dice (mDice): 0.7611



- # U-Net with ResNet-34 encoder
  - Residual connections improve gradient flow
  - Learns deeper semantic features
  - Better at capturing complex boundaries and shapes

Model Setup

- Architecture: UNet
- Encoder Backbone: ResNET-34
- Number of Classes: 4
  - Background
  - Vegetation
  - Farmland
  - Sand

Dataset

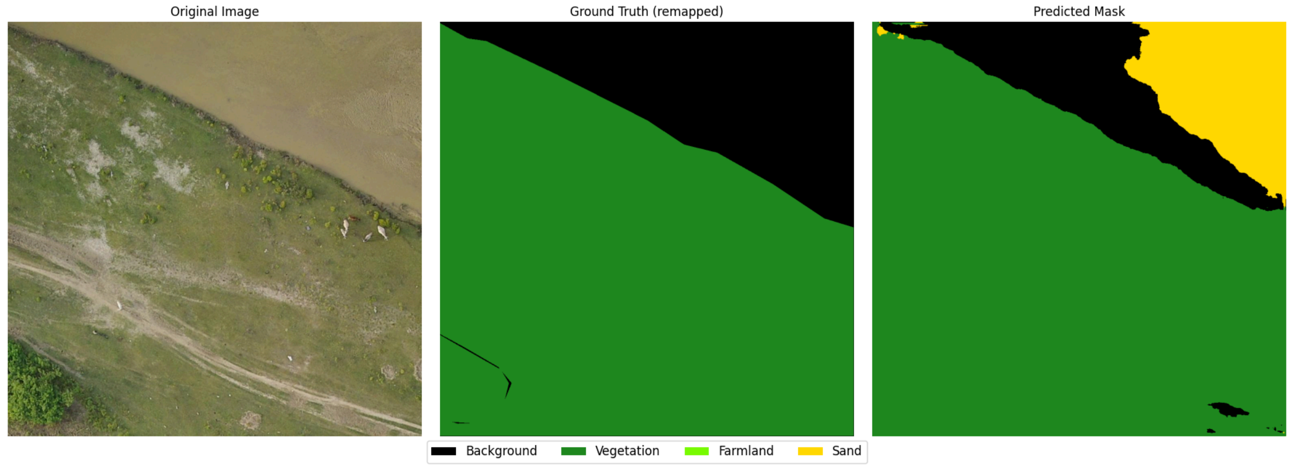
- Total Images: 771
  - Train: 698
  - Validation: 48
  - Test: 25

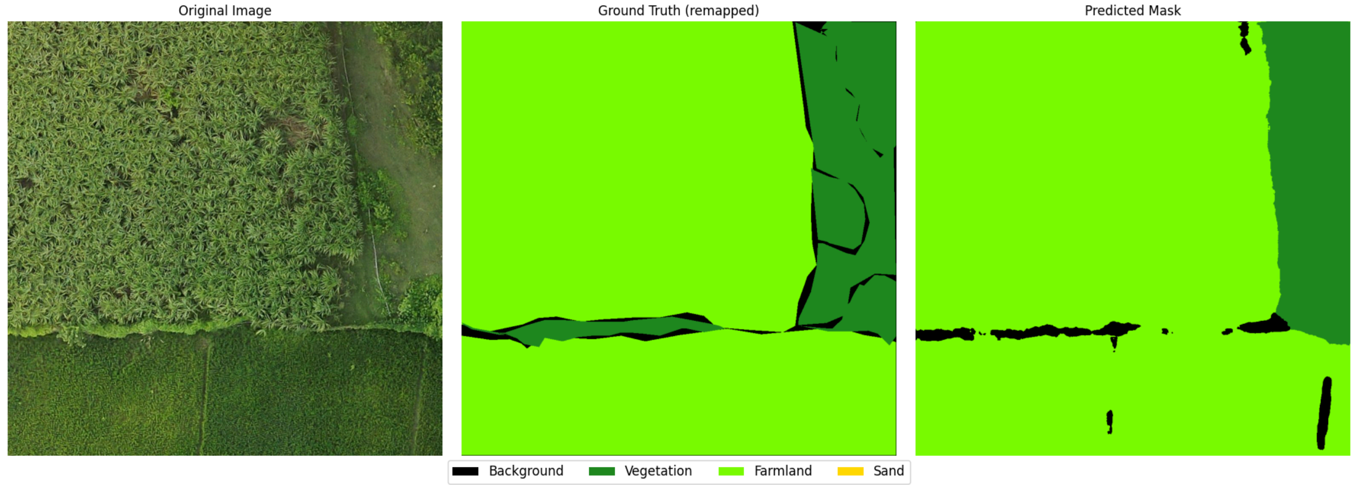
Class	IoU	Dice (F1)	Precision	Recall
Background	0.4298	0.6012	0.5369	0.6831
Barrenland	0.753	0.8591	0.9264	0.8009
Vegetation	0.4014	0.5729	0.8061	0.4444
Farmland	0.8898	0.9417	0.901	0.9863
Sand	0.7105	0.8307	0.851	0.8114

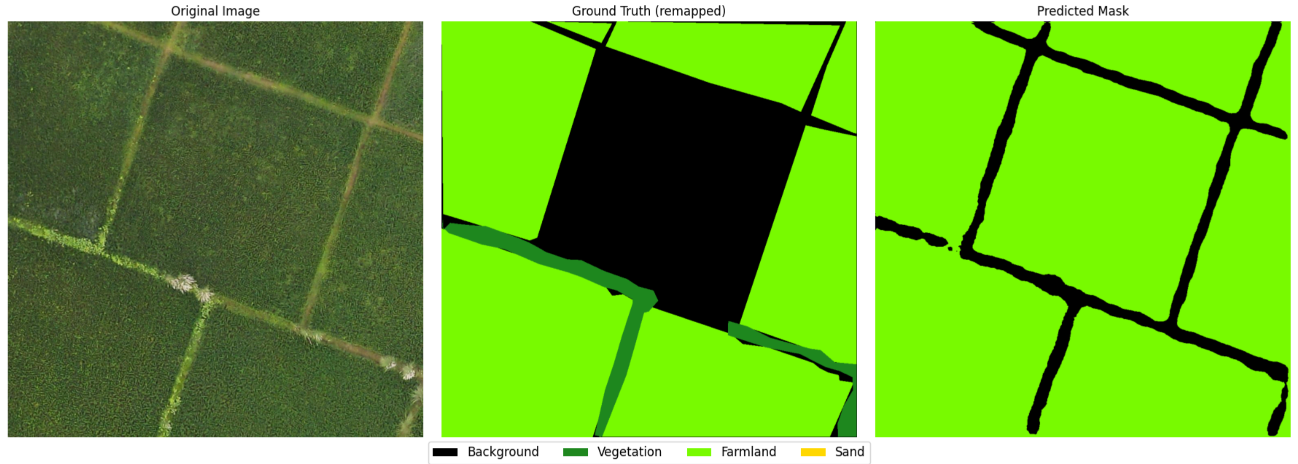
Overall Metrics

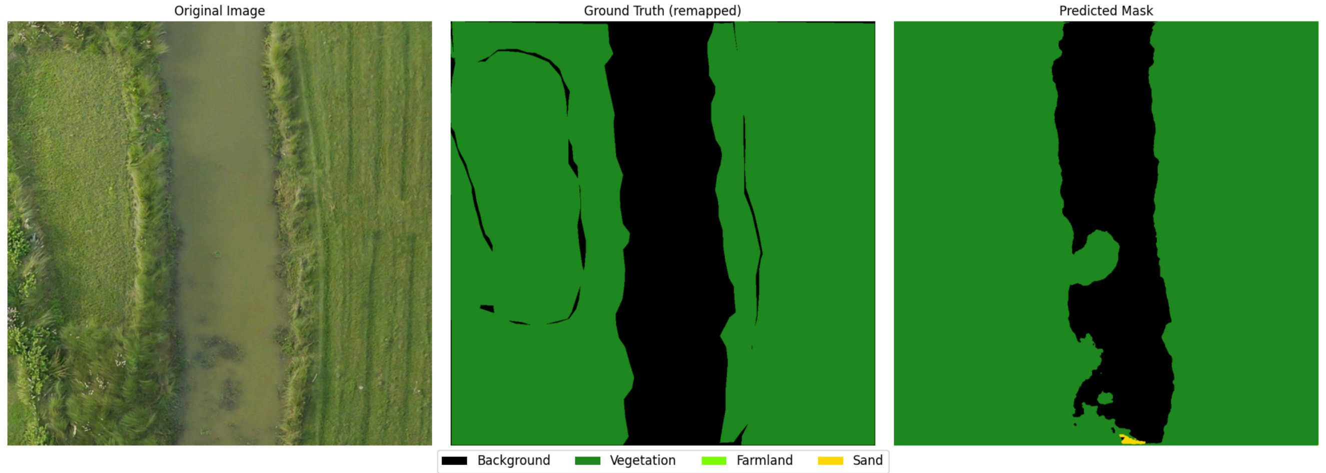
- Mean IoU (mIoU): 0.6769
- Mean Dice (mDice): 0.7711

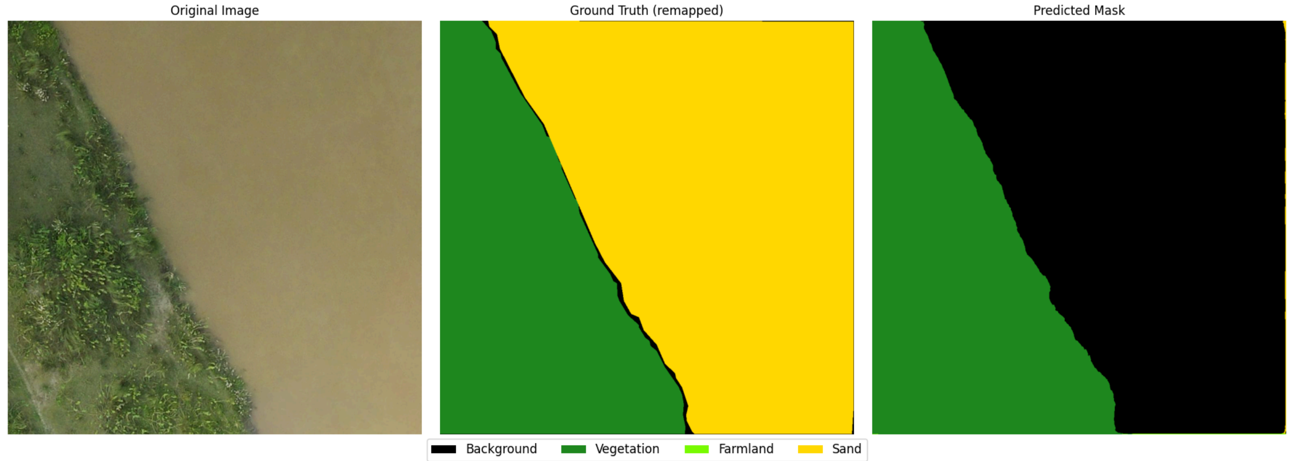
- **Samples prediction of Unet with RESNET34**

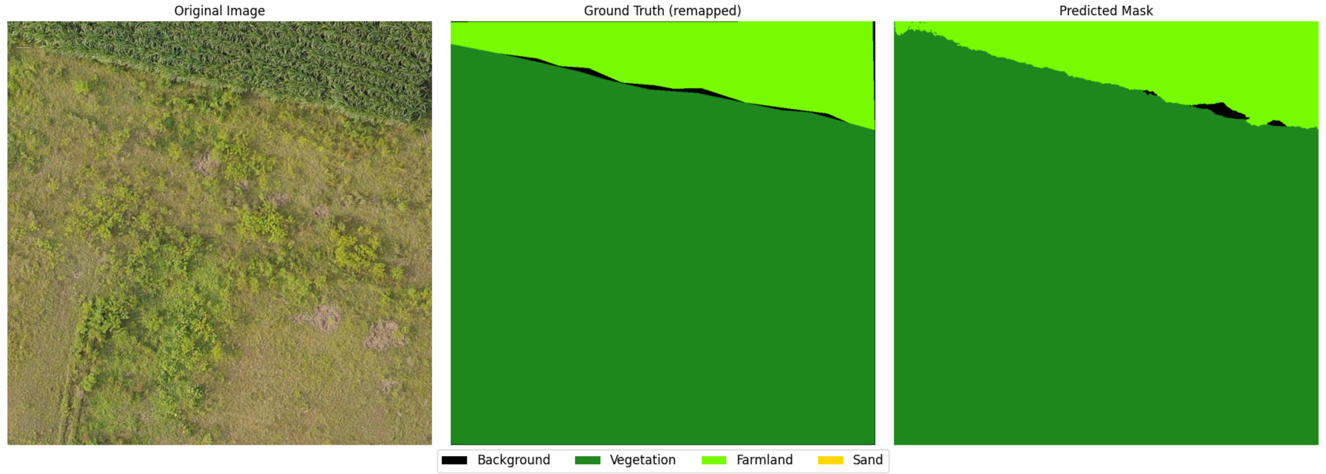




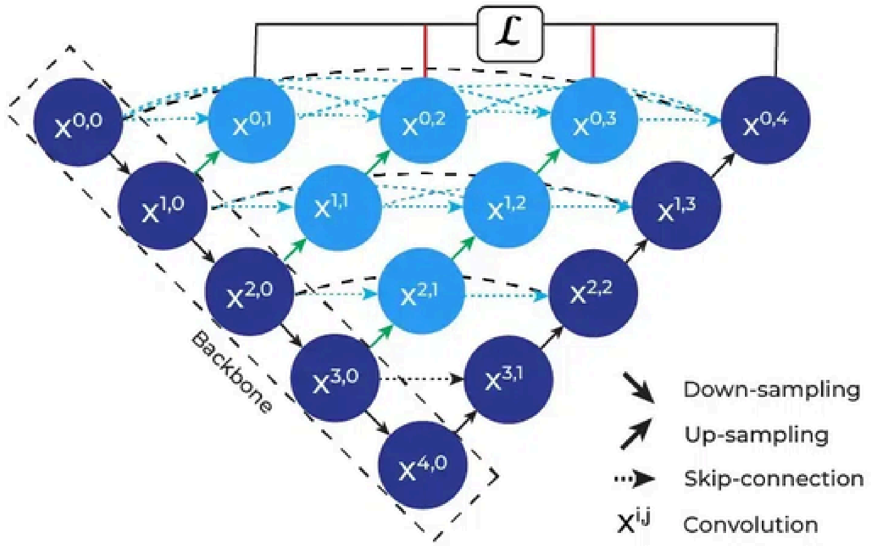








- **UNet++ with EfficientNet-B4**
  - UNet++ adds dense skip connections → better feature refinement.
  - EfficientNet-B4 provides high-quality multi-scale feature extraction with fewer parameters.



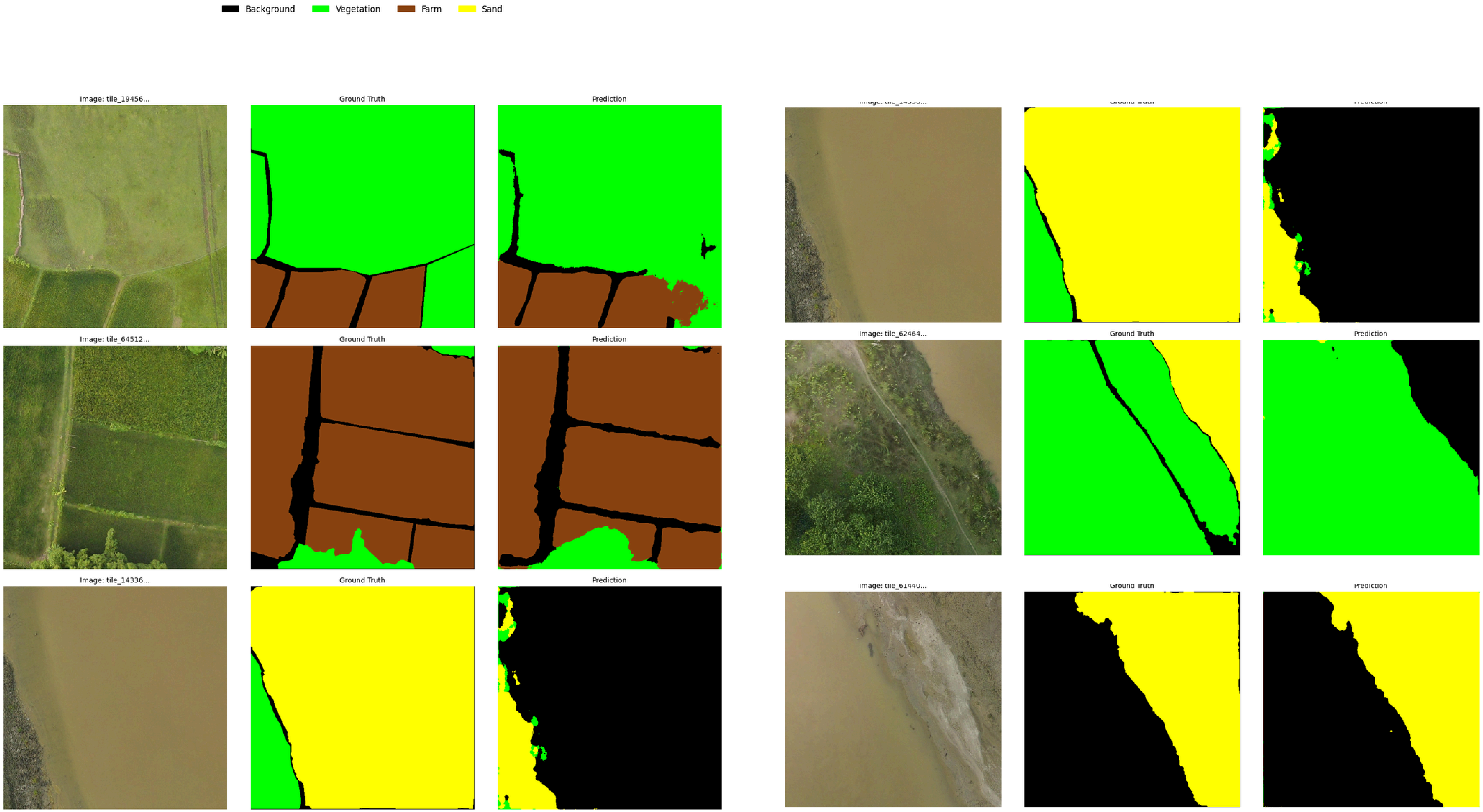
Model Setup
- Architecture: UNet++
- Encoder Backbone: EfficientNet-B4
- Number of Classes: 4
  - Background
  - Vegetation
  - Farmland
  - Sand

Dataset
- Total Images: 858
  - Train: 702
  - Validation: 101
  - Test: 55

Class	IoU	Dice/F1	Precision	Recall
Background	0.5967	0.7475	0.747	0.7479
Vegetation	0.7829	0.8782	0.8452	0.9139
Farmland	0.8345	0.9098	0.9364	0.8845
Sand	0.7931	0.8846	0.8918	0.8776

Overall Performance
- Mean IoU: 0.7518
- Mean Dice: 0.8550

- **Samples prediction of UNet++ with EfficientNet-B4**

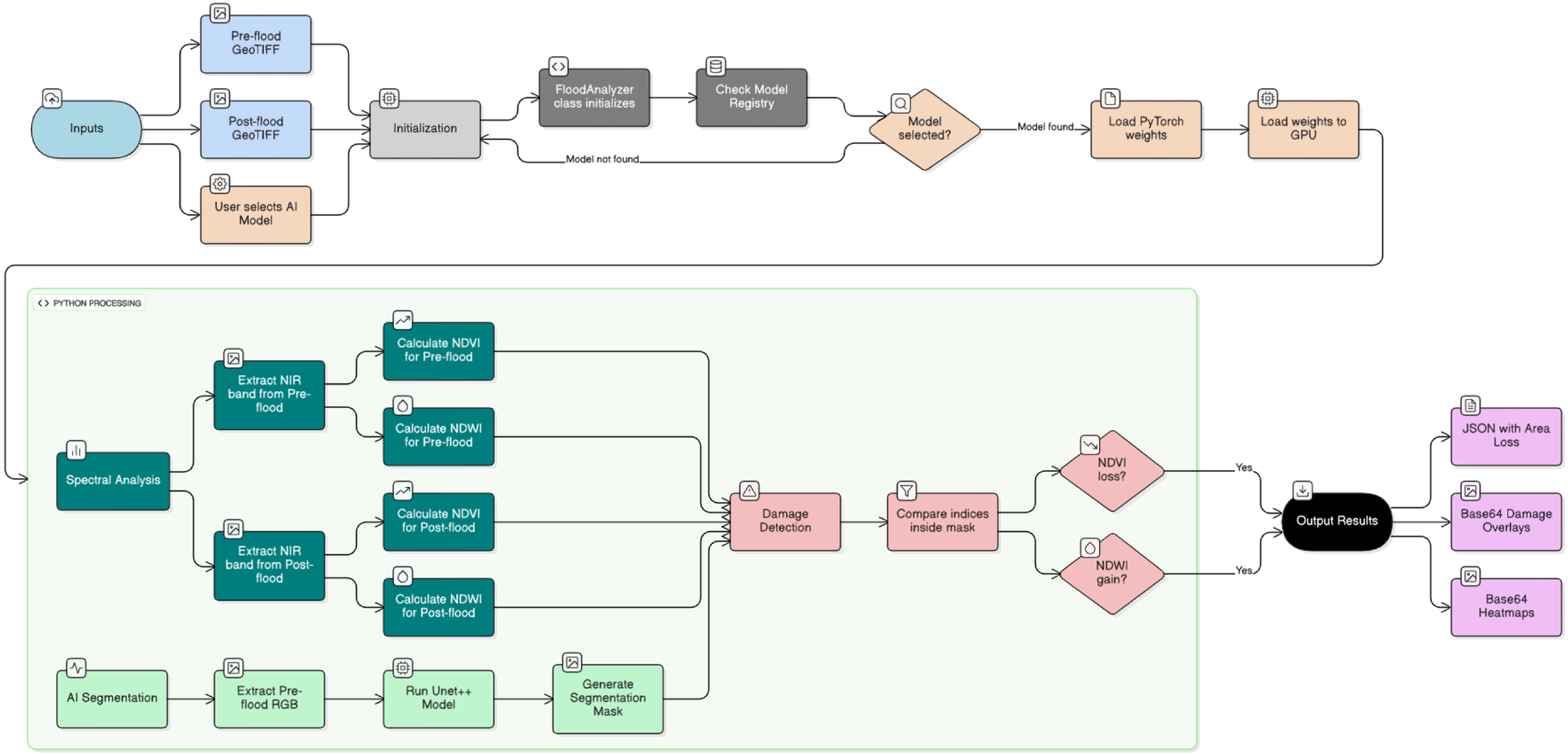


Comparison of All Models

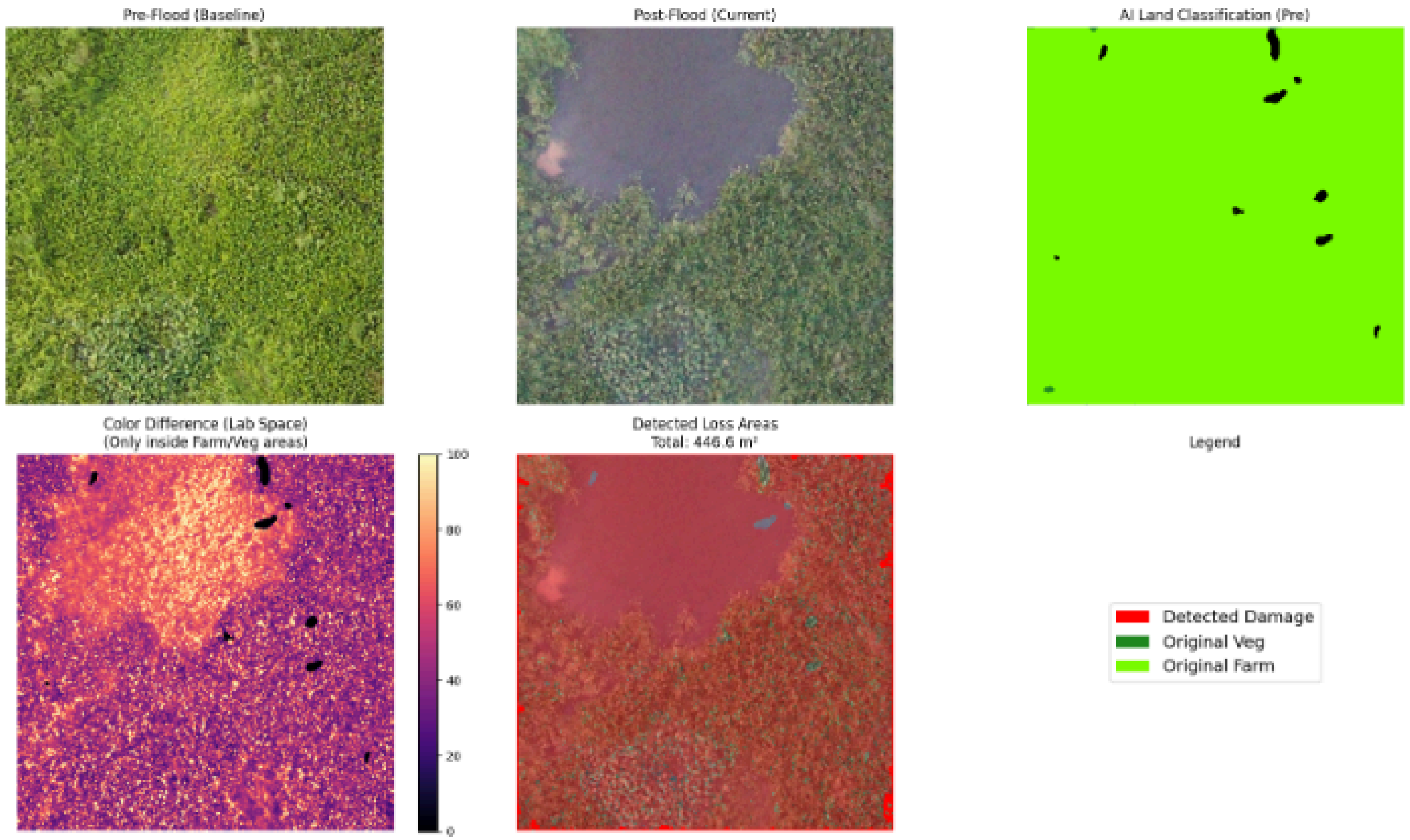
Summary Table of Model Performance

Model	Architecture	Encoder	Classes	Train/Val/Test	mIoU	mDice	Notes
Model 1	UNet	— (Basic UNet)	7	55 / 14 / —	*(Epoch 7) —*	Dice: **0.1121**	Very low performance due to small dataset + 7-class complexity
Model 2	UNet	**VGG16**	5	443 / 48 / 25	**0.6369**	**0.7611**	Strong improvement; good on Farmland & Barrenland
Model 3	UNet	**ResNet-34**	4	698/ 48 / 25	**0.6769**	**0.7911**	Similar results to UNet-VGG16; ResNet features helped stability
Model 4	**UNet++**	**EfficientNet-B4**	4	702 / 101 / 55	**0.7518**	**0.855**	**Best performing model**; very high scores across classes

System Design



- **Results and Output of the Loss assesment**



Farmland: 357.3 m² lost (out of 376.5 m²)

Vegetation: 105.9 m² lost (out of 111.0 m²)

Challenges

1. Working with Large GeoTIFF Orthomosaics

2. Tiling the Orthomosaics Correctly

3. Annotation Difficulties

4. Class Imbalance

5. Training Challenges

Future Enhancement

1. More Accurate and Consistent Annotations

2. Improved Training With Hyperparameter Tuning

3. Expanding and Diversifying the Dataset

4.Implement a More Reliable Method for Quantifying Actual Land Loss

Classes with their count

In total tile of 858 the details of each class are as shown below:

COLOR	CLASS NAME	COUNT
	Barrenland	718
	Bush	882
	edgecase	140
	Farmland	1,605
	Sand	268
	Tree	109

Team Details

Dharmendra Singh Chaudhary

BE-Computer Engineering (Student)

Contributions

- Model Research
- Model Training and Tuning
- Develop pipeline for inference and visualization

Shubham Thapa

Electronics and Communication and Information Engineer

Contributions

- Data Labeling & Mask Creation
- Model Architecture Selection
- Model Training and Tuning

Sangam Paudel

BE-Computer Engineering (Student)

Contributions

- Data Labeling & Mask Creation
- Quality Validation of Annotations

Susan Shakya

Contributions

- Frontend & UI Implementation
- Backend API Development & Integration

Thank you!

Any Queries ?