

Semantic Segmentation of Agricultural Field Patterns Using a Semi-supervised Deep Learning Approach

Bryce (Bo) Meyering

bmeyering3@gatech.edu

Francis Lin

flin96@gatech.edu

Stanislav Sheludko

ssheludko3@gatech.edu

Juanwen Lu

jlu435@gatech.edu

Abstract

Semantic segmentation of satellite imagery has significant implications for agricultural production from yield prediction to identifying total area of land mass where certain crops are grown. However, interpretation and annotation of satellite imagery is time-consuming, expensive, requires experienced remote sensing professionals, and represents a major hurdle to the development of segmentation models. Semi-supervised learning (SSL) methods offer a solution by combining labeled and unlabeled training data to iteratively improve performance. This project aims to apply SSL to field pattern recognition using both a fully labeled "Agriculture Vision" dataset and the newly released 2024 Agriculture Vision Dataset, comparing it with the vanilla, fully-supervised deep learning approach and the semi-supervised learning pseudo-label methods. The recent release of the unlabeled dataset and the ongoing 2024 challenge present an opportunity to push the boundaries of SSL in agricultural imagery analysis.

1. Introduction/Background/Motivation

Conventional row crop agricultural production relies on careful attention to and mitigation of field problems like weed outbreaks, nutrient deficiencies, and flooding to maximize crop yields, yet the scale of modern farming operations can make identifying areas needing corrective action challenging. Remote sensing, the use of satellite or unmanned aerial vehicle (UAV) RGB-NIR, multi-spectral, and hyperspectral imagery to segment certain types of objects, has revolutionized production by allowing producers to pinpoint these problems quickly to take corrective action [25].

Semantic segmentation modeling has been used in a variety of research fields from medicine to engineering to identify patterns in images corresponding to desired targets [15, 30, 8, 31], and has also found many applications in the agricultural sector from viticulture [4, 26], leaf disease segmentation [12], crop row detection for automated machinery [3], and weed/crop discrimination [28]. For field level remote sensing, the use of vegetative indices derived

from the RGB-NIR channels was long the dominant approach for detecting the boundaries of problem areas [9]. With the advent of better deep learning segmentation models and open-source data like the Agriculture Vision dataset [6], researchers have evaluated their efficacy in identifying problem patterns in a variety of agricultural remote sensing problems [18, 23, 20, 10].

However, the remote sensing image data needed to train large, deep learning segmentation models can be expensive to acquire and time-consuming to annotate, since the images need to be fully labeled at the pixel level [32, 17]. While training schemes like transfer learning and fine tuning work well to quickly spin up smaller prototype models, semi-supervised learning (SSL) is an umbrella term encompassing a variety of machine learning training schemes which make use of a small labeled dataset and abundant unlabeled data to generate "pseudo-labels" for the unlabeled data under self-training or mutual-training methods. The dataset is then extended with the "pseudo-labels" to train a more robust model with adding the loss term with the hyper parameter weighting the prediction loss from the unsupervised part [14, 34, 2, 33, 36]. SSL has been successfully applied in many different deep learning segmentation tasks [4, 12, 19]. This present project seeks to apply SSL techniques to the field pattern recognition by using the fully labeled "Agriculture Vision" dataset [6] in combination with the newly released unlabeled **2024 Agriculture Vision Dataset**. Semantic segmentation performance will be evaluated on the test set and compared with the vanilla, fully-supervised deep learning approach and the semi-supervised learning pseudo-label methods using both labeled and unlabeled datasets. Just released in 2024 challenge, the unlabeled field images provide us a rich opportunity to test the efficacy of SSL.

2. Approach

2.1. Dataset Preprocessing

The 2024 dataset includes 105Gb of new, unlabeled RGB-NIR photos, whereas the original dataset in 2021 is 20Gb and is fully annotated. For the labeled images, there

are a total of 9 classes including "double plant", "drydown", "endrow", "nutrient deficiency", "planter skip", "water", "waterway" and "weed cluster" and "background". The annotated dataset is split with a 60/20/20 train/validation/test ratio. Furthermore, there is a strong class imbalance in these labels with some categories occupying significantly larger portions of the label space. The images from the 2024 dataset were in large (9000x9000) 32bit float, tiff formatted files with one file for each of the 4 channels. We were provided with image mask for the entire image that would exclude non target areas from the images. To get the images in the same format as the labeled training set, we stacked all four channels together and then masked them to exclude non-target image features such as roads, buildings, etc. 32 bit images are notoriously noisy so they needed some pre-processing. Corrupted areas in the images with pixel values of -10000 were replaced with 0's, and then we clipped outlier pixels performing quantile clipping at the 3rd and 97th percentile of the pixel values per channel [6]. Finally all pixels were normalized to 0-255 and then the images were tiled to 512x512 JPEG images, resulting in 32,974 images.

2.2. Base Model and Backbone

Performance of semi-supervised method depends on two critical factors [33]. The first one is the selection of segmentation base model. Among the state-of-art segmentation models, we used DeepLabV3+[5], U-Net PlusPlus [35] and FPN[6]. We also tested these architectures with various encoder backbones such as ResNet34, ResNet101, and ResNext101.

2.3. Training Schemes with Pseudo Labels

We decided on a FixMatch pseudo-labeling approach to tackle this segmentation problem slightly modifying the original algorithm to remove the exponential moving average of model weights[27]. We drew from PyTorch implementations of FixMatch for image classification found in two GitHub repositories [7, 13], adapting and integrating their code to suit our semantic segmentation problem and data types. For the FixMatch method, briefly, we performed weak augmentations on the labeled training data and strong and weak augmentations on the unlabeled data. All of the images were passed through the model to compute the logits. Pseudo labels for the unlabeled portion of the data were calculated from the weakly augmented images and the unlabeled portion of the loss was calculated from the strongly augmented image logits with the pseudo-labels and pixels with pseudo label probabilities less than 0.8 were masked from loss contribution. This was scaled by a weighting factor λ with the supervised portion of the loss.

For pseudo label prediction, we also explore the effectiveness of few-shot learning concept on the segmentation of agricultural field images. Five labeled images from each

classes respectively were selected as the support set and as the query set for evaluation. Feature vectors of the images from support set and query set were extracted using 3 different pretrained models including ResNet18, ResNet50 and ResNet101. The predicted class in the query set is based on the distance between its feature vector and the feature vectors of other labeled images in each classes. Distance calculation in MatchingNet and ProtoNet were considered for comparison. If the approach of few-shot learning performed well on accuracy, we would propose to update the pseudo labels for the images with low confidence level during SSL.

2.4. Loss Functions

2.4.1 Class Imbalance

The choice of an appropriate loss function was critical due to severe class imbalance in the 2021 training dataset. The distribution in class categories is depicted in Figure 1 on a pixel and image basis. Several different loss functions were considered to address the class imbalance problem.

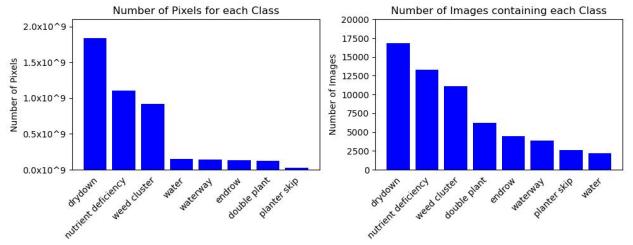


Figure 1. The 2021 training dataset exhibits severe class imbalance on both a pixel and image level.

We evaluated several different loss functions with our models to determine the best one in combination with different optimizer hyperparameters.

2.4.2 Weighted Cross Entropy

Initially we started with a simple class weighted multiclass cross entropy loss which comes prepackaged in Pytorch. This takes weights w_c for each discrete label class (generally inverse incidence weights) and one-hot encoded labels per pixel, y_c to return a loss value that places more weight on underrepresented sample classes.

$$CE_w = -\frac{1}{N} \sum_{n=1}^N w_c \cdot y_c \cdot \log(\hat{y}_c) \quad (1)$$

2.4.3 Dice Loss

Generalized Dice Loss was introduced in 2017 to overcome some shortcomings of standard cross entropy [29]. For a given class $c \in C$, the loss can be defined as

$$DICE_c = 2 \frac{\sum_i (S_{ci} G_{ci}) + \epsilon}{\sum_i (S_{ci}) + \sum_i (G_{ci}) + \epsilon} \quad (2)$$

Where S_{ci} is the softmax output for class c at the i^{th} pixel and G_{ci} is the ground truth for that pixel, and ϵ is a small value added to avoid division by zero. These scores for each class are then aggregated as

$$\mathcal{L}_{dice} = 1 - \frac{1}{C} \sum_{c=1}^C DICE_c \quad (3)$$

Dice loss has some advantage over cross entropy in that it directly optimizes our evaluation criterion IOU. However, it can also lead to some gradient problems if either G_{ci} or S_{ci} are very small. Adding the ϵ helps stabilize this to some degree.

2.4.4 Tversky Loss

Tversky loss was also introduced in 2017 to tackle class imbalance problems in semantic segmentation problems and allows for some additional finetuning of the loss function to weight more heavily for false positives and false negatives [22]. For class $c \in C$ we define the class loss as

$$Tversky_c = \frac{\sum_i (S_{ci} \cdot G_{ci}) + \epsilon}{\sum_i (S_{ci} \cdot G_{ci}) + \alpha FP + \beta FN + \epsilon} \quad (4)$$

where S_{ci} are the pixelwise softmax output for class c at pixel i , FP is the false positive pixels defined as

$\sum_i (S_{ci} \cdot (1 - G_{ci}))$ and FN are the false negatives defined as $\sum_i ((1 - S_{ci}) \cdot G_{ci})$. Each class loss is aggregated into the multiclass loss as

$$\mathcal{L}_{Tversky} = 1 - \frac{1}{C} \sum_{c=1}^C Tversky_c \quad (5)$$

Tversky loss is unique in that you can specifically control how to weight the false negative and false positive classifications to finetune the model loss. When $\alpha = \beta = 0.5$ the loss can be shown to be a generalization of Dice Loss [22].

2.4.5 Focal Tversky Loss

A focal version of Tversky loss published in 2018 [1] builds upon the previously defined Tversky index for class c (equation 4) by adding a focal parameter γ to change the contribution of a given images loss contribution in a non-linear fashion.

$$\mathcal{L}_{FT_c} = \sum_{c=1}^C (1 - Tversky_c)^{\frac{1}{\gamma}} \quad (6)$$

In practice this loss is averaged over all classes C in a mini-batch to get

$$\mathcal{L}_{FT} = \frac{1}{C} \sum_{c=1}^C (1 - Tversky_c)^{\frac{1}{\gamma}} \quad (7)$$

The focal modification on the Tversky loss helps to focus the model training on those examples that are difficult to classify and suppresses the contribution of easy to classify examples from the loss function, without having specific class weighted contributions.

2.4.6 Adaptive Class Weighted Loss

The authors of one of the leaderboard solutions to the 2020 Agriculture Vision Competition [16] implemented a novel approach on loss calculation method called Adaptive Class Weighted Loss (ACW Loss). ACW Loss is designed to address the imbalanced problem of the images in which most pixels are under the class of background and only some of the pixels belong to the target class we hope to identify.

$$\mathcal{L}_{acw} = \frac{1}{|Y|} \sum_{i \in Y} \sum_{j \in C} \tilde{w}_{ij} * p_{ij} - \log(\text{MEAN}\{d_j | j \in C\}), \quad (8)$$

where Y contains all the labeled pixels, \tilde{w}_{ij} is the iterative weights to pixel level and the dice coefficient d_j is given as

$$d_j = \frac{2 \sum_{i \in Y} y_{ij} \tilde{y}_{ij}}{\sum_{i \in Y} y_{ij} + \sum_{i \in Y} \tilde{y}_{ij}} \quad (9)$$

where $\tilde{y}_{ij} \in (0, 1)$ and $y_{ij} \in (0, 1)$ denote the ij -th prediction and the ground-truth of class j separately in the current training images.

2.5 Optimizer

We employed two different common optimizers to fine tune all of our model tuning experiments.

2.5.1 Stochastic Gradient Descent

Stochastic Gradient Descent(SGD) is a traditional and effective optimization approach to update the parameters based on the output changes of the predefined loss function. The learning rate in the equation of weight updates controls the step size in the direction of gradient. η is the learning rate. Our implementation used Nesterov momentum in the update step.

2.5.2 ADAM

Another popular stochastic optimization method is *Adam* (Adaptive Moment Estimation) which requires first-order

gradient only and it delivers strong optimization performance. Adam works by setting an adaptive per parameter learning rate based on the first and second moments of the parameters in addition to an exponentially decayed moving average of the parameters from past optimizer iterations. These changes allow for ADAM to handle sparse gradient updates better than more common vanilla optimizers like SGD.

2.6. Project Scope, Outlook, and Approach Changes

Both sets of fully supervised and semi-supervised trained models will be evaluated on the test holdout set using the mean Intersection-over-Union (mIoU)[6, 32]:

$$mIoU = \frac{1}{c} \sum \frac{\text{Area}(P_c \cap T_c)}{\text{Area}(P_c \cup T_c)} \quad (10)$$

where c is the number of classes, P_c is the predicted mask of class c , and T_c is the ground truth mask of class c .

However, due to the nature of the ongoing competition, the targets for the test set were reserved from this year dataset since the final models are evaluated on that. Thus for the supervised data we combined the existing training and validation sets, stratified based on the majority label of each image (Figure 1), and resampled a new train, validation, and test set in a 60/20/20 split. With all the images combined we calculated the majority class label for each image mask and stratified the data.

While we fully anticipated long training times and computational challenges when applying FixMatch on a complex image dataset, we did not expect optimization and convergence problems and were ultimately unable to overcome them. None of the approaches that we originally planned to use ended up working out, leading us to pivot to a simpler approach. We ended up taking some parts of our previous approach and combining them with Team SCG Vision’s approach for the 2020 Agriculture Vision Competition [24]. Their solution to the fully supervised problem included a self-constructing graph module in the network with multiple images augmentations which can learn latent image features to construct segmentation maps. While we retrained their model on our reshuffled dataset to use as a benchmark, we ultimately had better performance from DeepLabV3+ as explained below. However, one of the most important adaptations we used from their codebase was their implementation of an adaptive class weighted (ACW) loss function which outperformed the other loss functions tested.

2.7. Development Environment

All models and code were developed in Python 3.10 using a Pytorch 2.2 framework. We used the popular library ‘segmentation_models.pytorch’ [21] which builds off of ‘timm’ [11] encoders for most of the model architectures tried. Model development and prototyping was performed

on an NVIDIA RTX 4060 8GB in Ubuntu. Full model training was performed on a commercial workstation with 4x40GB A100 GPUS run in parallel.

3. Experiments and Results

3.1. FixMatch Implementation

Our original approach focused on using a modified version of the FixMatch algorithm [27] to train a model on the labeled and unlabeled training data. We measured success based on the overall mIOU criterion. Our goal was to train a model achieving benchmark performance of 0.5 to 0.63, typical for this dataset across different competition years. However, we encountered tremendous difficulty optimizing the model and stabilizing the loss even over a wide range of tested hyperparameters. The loss values decreased rapidly from ≈ 2 to 1, and then fluctuated erratically from 0.5 to 1.5, with no more improvement after the first epoch. Mean IOU values from these models were very low, in the 0.05 to 0.1 range, indicating that the model was not able to optimize the feature space appropriately. We tested out the different loss functions mentioned above with the FixMatch implementation and all of them failed to converge with either Adam or SGD (see Figure 2). The Tversky losses in particular were prone to exploding or vanishing gradient problems over a wide range of hyperparameters for SGD and ADAM optimizers.

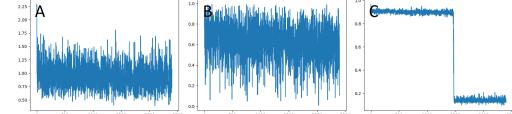


Figure 2. A) Cross Entropy Loss, B) Dice Loss, C) Tversky Loss

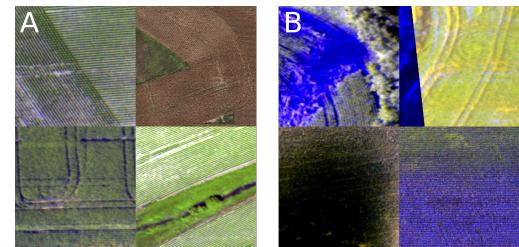


Figure 3. Image samples from the 2021 training dataset. A) Samples which clearly show the target field patterns with the correct colors. B) Samples with imaging artifacts due to color imbalance or errors in acquisition.

One possible reason why FixMatch was unable to converge is because FixMatch is extremely sensitive to batch size. FixMatch utilizes unlabeled data to generate pseudo-labels, which are then incorporated into the training process alongside labeled data. With smaller batch sizes, the diversity of unlabeled samples included in each batch de-

creases. Consequently, the reliability of pseudo-labels generated from these batches diminishes, potentially introducing more noise into the training process due to imprecise gradient estimates. Because batch size is limited by computational resources, we were unable to mitigate this issue when training locally on consumer GPU hardware.

Another reason is the variability in image quality, as shown in Figure 3. While many images, as shown in Figure 3A, have appropriate color balance and contrast, there are a good amount of images where the RGB channels seem to be out of sync with each other, perhaps due to noise pixels in the original, or if the normalization was applied channel wise (inset 3B). We noticed similar artifacts when normalizing and tiling the 2024 unlabeled data if the quantile clipping was too conservative (1-99th clipping) or too aggressive (5-95th percentile). Additionally it was common to find images whose RGB channels appeared to be improperly stacked, giving the images a strange appearance. These image quality issues pose a significant challenge to optimizing the model, since any additional augmentation of the images during FixMatch results in an even lower quality image during model training.

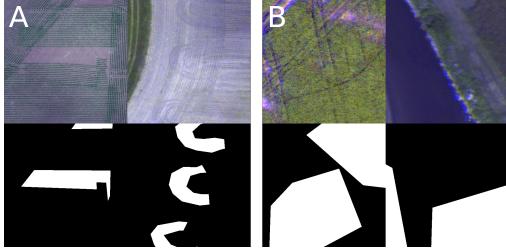


Figure 4. RGB images and mask samples from the 2021 training dataset. A) Sample masks which correspond clearly to target field patterns for classes (L to R) 'planter-skip' and 'endrow'. B) Sample masks do not correspond well to noticeable field patterns for classes 'nutrient-deficiency' and 'drydown'.

Furthermore, examining the labels revealed varying levels of clarity regarding the corresponding features in each image. For some classes such as 'planter-skip' where the tractor missed some planting rows or 'endrow' where the field equipment turned around at the end of a pass, the masks correspond well to the field pattern, as shown in Figure 4A. However, for many classes like 'nutrient-deficiency' and 'drydown' pictured in Figure 4B, the masks lack discernible patterns, leading to uncertainty about what the model actually learns. This issue is compounded when a class is underrepresented in the dataset.

3.2. SSL Method Exploration: Few-shot Learning

To test the prediction performance of few-shot learning method, we implemented 9-way 5-shot learning approach under three selected pretrained models for feature extraction. And the similarity for classification is estimated using

both cosine and euclidean distance method. As shown in supplementary figure 1, the best accuracy was obtained by the most complicated model using cosine similarity, however the accuracy was still lower than 33% and each of the classes vary in accuracy due to not only the imbalanced problem on pixel level and on image level but also the absence of more fitted pretrained models to extract the features from agricultural patterns. Therefore, we did not pursue the few-shot learning method any further for this project.

3.3. Final Approach and Results

To benchmark performance of our models we selected the 2020 Agrivision competition top-3 leaderboard model proposed in [16]. To train this MSCG-Net model we followed the steps described in [24], however the code had to be adapted to work in a Pytorch 2 framework on the 2021 competition dataset, which has 9 classes vs. 7 classes in the 2020 competition dataset. The authors in [16] report mIoU of **0.509** for MSCG-Net with Se_Resnext101 backbone after training on single GPU for 25 epochs with batch size of 10. Utilizing 4x40GB GPUs, we were able to increase the batch size to 40 and train the benchmark MSCG-Net model for 70 epochs and achieve the mIoU of **0.565** on the 2021 competition test set. Conventional multi-parameter grid search incurred high computational cost, therefore experiments were limited to 1 epoch randomized grid search. Se_Resnext101 was chosen over Resnet50, Resnet101, Se_Resnext50 as the encoder backbone for the main DeepLabV3+ model; ACW loss was selected over vanilla Crossentropy loss, Dice Loss and class-balanced Focal Loss and SGD optimizer was selected over Adam as the result of these initial experiments. Using ACW loss proved advantageous over other forms of loss used to address class imbalance. Instead of relying on fixed weights computed once for the entire dataset, the adaptive approach adjusts class weights iteratively, allowing the model to respond to changing data distributions. This mechanism allows ACW to capture local variations within batches, leading to more precise adjustments. This is particularly pertinent to our AgriVision dataset, given its highly variable image content. Batch sizes of 40 and 100 were also compared and larger batch size displaying deteriorating performance, which was probably due to gradients updating less frequently.

We also trained a fully supervised baseline DeepLabV3+ network with a ResNext101 32x4d encoder backbone for 20 epochs with batch size 40 and ACW loss and SGD optimization. This network converged quickly at 20 epochs and outperformed the benchmark MSCG-Net model on the test set achieving the mIoU of **0.609** (Table 1, supp. fig. 2). This model's predictions were very reasonable in comparison to the ground truth (figure 5) even for different classes such as 'drydown'. The class mIoU scores were revealing as well. Classes such as 'endrow' which are easy to visualize, had

the lowest score, while the 'drydown' score was higher than the overall mIoU. This suggests that there might be more features, especially in the NIR channel, which contribute to detecting these patterns that are difficult for the human eye to observe directly. The mean, pixel classification accuracy was **0.826** and the overall model F1 score was **0.749**.

Class	Supervised mIoU	SSL mIoU
Background	0.770	0.630
Double Plant	0.476	0.225
Drydown	0.670	0.600
Endrow	0.374	0.186
Nutrient Deficiency	0.508	0.388
Planter Skip	0.665	0.303
Water	0.793	0.322
Waterway	0.681	0.447
Weed Cluster	0.548	0.349
mIoU	0.609	0.383

Table 1. Class IoU and mIoU for the two DeepLabV3+ models trained.

The supervised DeepLabV3+ model epoch 20 checkpoint was used to generate pseudo labels for each image in the 2024 dataset. A confidence threshold of 0.6 was applied to each of the prediction tensors to discard pixels with confidence less than 0.6. Predictions with no confident pixels left were discarded in entirety, removing approximately 11,000 images. This pseudo-labeled set was then combined with the original 2021 training set, with pseudo-labels used as the ground truth images. The same transformations were applied as for the original 2021 train set at this point. The DeepLabV3+ network with the same settings as the baseline model described above was then re-trained with batch size of 40 for 100 epochs and tested against the original test set achieving the mean IoU of **0.383** on test set and a classification accuracy of **0.721**.

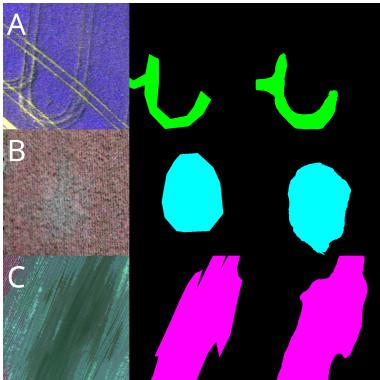


Figure 5. Test set image (RGB and NIR) inference from the fully supervised DeepLabV3+ model. From L to R, original image, ground truth target, model predictions. A) 'Endrow' B) 'Drydown' C) 'Water'

We have some ideas as to why our SSL model perfor-

mance was lower than the fully supervised model. First, the supervised model had already found a very good minimum, but we started training the SSL model from scratch, not from the supervised model weights. This could have led to much faster convergence since any gains would have been additive to what the supervised model had already learned. Second, we used a confidence threshold 0.6 for the SSL model which is lower than what we used for our Fixmatch implementation, 0.8. We thought that by allowing more of the lower confidence predictions in as pseudo labels, the model would learn more of the less confident classes better than the supervised DeepLab model had, which was not the case. Third, the SSL model was trained for 102 epochs total but appeared to be stuck in a difficult local minima. This is evidenced by the sharp decrease in the loss at epoch 101-102 which had basically plateaued at epoch 25. Further, sup. figure 3 also shows an uptick in mIOU and accuracy at epoch 88. We found that using vanilla SGD for DeepLab gave us better initial performance on the fully supervised set, but it is possible that SGD got stuck in a saddle point where a more robust optimizer would have been able to break out earlier. Finally, we opted to hard-code the pseudo labels based on the supervised model's parameter state at epoch 20. We could have adapted the approach to include dynamically updated pseudo-labels every batch, except without the strong augmentations to see if that improved performance. By making the pseudo labels static, we essentially enforced the assumption that the model made correct predictions at epoch 20 and would not learn the features better in an iterative fashion.

4. Conclusion and Future Work

This project proved to be a difficult challenge in terms of selecting the proper loss functions and optimizers, but we were able to train a DeepLabV3+ model using ACW loss which outperformed some of the recent benchmarks on the supervised training set. One avenue for future work is to revisit our FixMatch implementation with increased computational resources. We suspect that many of our model experiments failed to converge due to high data variability within a small batch size. Increasing computational resources would enable us to experiment with larger batch sizes, potentially stabilizing and reducing loss. Training a large model on a large dataset can be very expensive and time-consuming and it was difficult to quickly run experiments to determine what worked and what didn't. Another direction for future work is to experiment with using different ratios of labeled or unlabeled training data to determine the lower limit of labeled data needed to train a decent model on this dataset.

5. Work Division

Summary of contributions are provided in sup. Table 2.

Student Name	Contributed Aspects and Details
Francis Lin	Metrics, validation, visualization
Juanwen Lu	SSL methods, prototype model development
Bryce (Bo) Meyering	Fixmatch, loss functions, dataset processing
Stanislav Sheludko	Model training, code adaptation, SSL training

Table 2. Contributions of team members.

References

- [1] Nabil Abraham and Naimul Mefraz Khan. A NOVEL FOCAL TVERSKY LOSS FUNCTION WITH IMPROVED ATTENTION U-NET FOR LESION SEGMENTATION. 2018. 3
- [2] Willian Paraguassu Amorim, Everton Castelão Tetila, Hemerson Pistori, and João Paulo Papa. Semi-supervised learning with convolutional neural networks for UAV images automatic recognition. *Computers and Electronics in Agriculture*, 164:104932, 9 2019. 1
- [3] Maoyong Cao, Fangfang Tang, Peng Ji, and Fengying Ma. Improved Real-Time Semantic Segmentation Network Model for Crop Vision Navigation Line Detection. *Frontiers in Plant Science*, 13:898131, 6 2022. 1
- [4] A Casado-García, · J Heras, · A Milella, and · R Marani. Semi-supervised deep learning and low-cost cameras for the semantic segmentation of natural images in viticulture. 23:2001–2026, 2022. 1
- [5] Liang Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11211 LNCS:833–851, 2 2018. 2
- [6] Mang Tik Chiu, Xingjian Xu, Yunchao Wei, Zilong Huang, Alexander Schwing, Robert Brunner, Hrant Khachatrian, Hovnatan Karapetyan, Ivan Dozier, Greg Rose, David Wilson, Adrian Tudor, Naira Hovakimyan, Thomas S Huang, and Honghui Shi. Agriculture-Vision: A Large Aerial Image Database for Agricultural Pattern Analysis. In *CVPR*, 2020. 1, 2, 4
- [7] fbuchert. PyTorch Implementation: FixMatch, 2021. 2
- [8] Enhui Gu, Gang Xiao, Faming Lian, Tongyao Mu, Jie Hong, and Jun Liu. Segmentation and Evaluation of Crack Image From Aircraft Fuel Tank via Atrous Spatial Pyramid Fusion and Hybrid Attention Network. *IEEE Transactions on Instrumentation and Measurement*, 72, 2023. 1
- [9] Jerry L Hatfield, John H Rueger, Thomas J Sauer, Christian Dold, Peter O'brien, and Ken Wacha. Applications of Vegetative Indices from Remote Sensing to Agriculture: Past and Future. *Inventions*, 4(4)(71), 2019. 1
- [10] Qingqing Hong, Yue Zhu, Wei Liu, Tianyu Ren, Changrong Shi, Zhixin Lu, Yunqin Yang, Ruiting Deng, Jing Qian, and Changwei Tan. A segmentation network for farmland ridge based on encoder-decoder architecture in combined with strip pooling module and ASPP. *Frontiers in Plant Science*, 15:1328075, 2 2024. 1
- [11] Hugging Face and Ross Wightman. pytorch-image-models, 2024. 4
- [12] So-Yeon Jang, Goo-Young Moon, and Jong-Ok Kim. Enhancing Plant and Disease Segmentation through Semi-Supervised Learning with Feature Distillation. In *IEEE International Conference on Consumer Electronics-Asia*, 2023. 1
- [13] kekmodel. FixMatch, 2020. 2
- [14] Dong-Hyun Lee. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. 1
- [15] Xianan Li, Lecheng Jia, Fengyu Lin, Fan Chai, Tao Liu, Wei Zhang, Ziquan Wei, Weiqi Xiong, Hua Li, Min Zhang, and Yi Wang. Semi-supervised auto-segmentation method for pelvic organ-at-risk in magnetic resonance images based on deep-learning. *Journal of applied clinical medical physics*, 25(3), 3 2024. 1
- [16] Qinghui Liu, Michael Kampffmeyer, Robert Jenssen, and Arnt-Børre Salberg. Multi-view Self-Constructing Graph Convolutional Networks with Adaptive Class Weighting Loss for Semantic Segmentation. 3, 5
- [17] Zifei Luo, Wenzhu Yang, Yunfeng Yuan, Ruru Gou, and Xiaonan Li. Semantic segmentation of agricultural images: A survey. *Information Processing in Agriculture*, 2 2023. 1
- [18] Shruti Nair, Sara Sharifzadeh, and Vasile Palade. Farmland Segmentation in Landsat 8 Satellite Images Using Deep Learning and Conditional Generative Adversarial Networks. *Remote Sensing 2024, Vol. 16, Page 823*, 16(5):823, 2 2024. 1
- [19] Artzai Picon, Itziar Eguskiza, Pablo Galan, Laura Gomez-Zamanillo, Javier Romero, Christian Klukas, Arantza Bereciartua-Perez, Mike Scharner, and Ramon Navarra-Mestre. Improving Disease Quantification in Agricultural Imagery: Multi-Crop Neural Networks for Semantic Segmentation with Contextual Metadata and Semi-Supervised Learning. *Artificial Intelligences in Agriculture*, 2023. 1
- [20] Liang Qi, Danfeng Zuo, Yirong Wang, Ye Tao, Runkang Tang, Jiayu Shi, Jiajun Gong, and Bangyu Li. Convolutional Neural Network-Based Method for Agriculture Plot Segmentation in Remote Sensing Images. *Remote Sensing 2024, Vol. 16, Page 346*, 16(2):346, 1 2024. 1
- [21] qubvel. Segmentation Models Pytorch, 2023. 4

- [22] Seyed Sadegh, Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3D fully convolutional deep networks. 2017. 3
- [23] Furkat Safarov, Kuchkorov Temurbek, Djumanov Jamoljon, Ochilov Temur, Jean Chamberlain Chedjou, Akmalbek Bobomirzaevich Abdusalomov, and Young Im Cho. Improved Agricultural Field Segmentation in Satellite Imagery Using TL-ResUNet Architecture. *Sensors (Basel, Switzerland)*, 22(24), 12 2022. 1
- [24] samleoqh. MSCG-Net, 2020. 4, 5
- [25] Rajendra P Sishodia, Ram L Ray, and Sudhir K Singh. Applications of Remote Sensing in Precision Agriculture: A Review. *Remote Sens.*, 2020. 1
- [26] Petar Slavićek, Ivan Hrabar, and Zdenko Kovačić. Generating a Dataset for Semantic Segmentation of Vine Trunks in Vineyards Using Semi-Supervised Learning and Object Detection. *Robotics*, 13(2):20, 1 2024. 1
- [27] Kihyuk Sohn, David Berthelot, Chun Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. *Advances in Neural Information Processing Systems*, 2020-December, 1 2020. 2, 4
- [28] Daniel Steininger, Andreas Trondl, Gerardus Croonen, Julia Simon, and Verena Widhalm. The CropAndWeed Dataset: a Multi-Modal Learning Approach for Efficient Crop and Weed Manipulation. In *Winter Conference on Applications of Computer Vision (WACV) 2023*, 2023. 1
- [29] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M Jorge Cardoso. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. 2
- [30] Shaheen Syed, Kathryn E Anderssen, Svein Kristian Stormo, and Mathias Kranz. Weakly supervised semantic segmentation for MRI: exploring the advantages and disadvantages of class activation maps for biological image segmentation with soft boundaries. *Scientific Reports* —, 13:2574, 2023. 1
- [31] Jingjing Tao, Zhe Chen, Zhongchang Sun, Huadong Guo, Bo Leng, Zhengbo Yu, Yanli Wang, Ziqiong He, Xiangqi Lei, and Jinpei Yang. Seg-Road: A Segmentation Network for Road Extraction Based on Transformer and CNN with Connectivity Structures. *Remote Sensing* 2023, Vol. 15, Page 1602, 15(6):1602, 3 2023. 1
- [32] Mang Tik Chiu, Xingqian Xu, Kai Wang, Jennifer Hobbs, Naira Hovakimyan, Thomas S Huang, Honghui Shi, Yunchao Wei, Zilong Huang, Alexander Schwing, Robert Brunner, Ivan Dozier, Wyatt Dozier, Karen Ghandilyan, David Wilson, Hyunseong Park, Junhee Kim, Sungho Kim, Qinghui Liu, Michael C Kampffmeyer, Robert Jenssen, Arnt B Salberg, Alexandre Barbosa, Rodrigo Trevisan, Bingchen Zhao, Shaozuo Yu, Siwei Yang, Yin Wang, Hao Sheng, Xiao Chen, Jingyi Su, Ram Rajagopal, Andrew Ng, Van Thong Huynh, Soo-Hyung Kim, In-Seop Na, Ujjwal Baid, Shubham Innani, Prasad Duttande, Bhakti Baheti, Sanjay Talbar, and Jianyu Tang. The 1st Agriculture-Vision Challenge: Methods and Results. In *CVPR*, 2020. 1, 4
- [33] Jesper E van Engelen, Holger H Hoos, Tom B Fawcett Jesper E van Engelen, and Holger H Hoos hh. A survey on semi-supervised learning. *Machine Learning*, 109:373–440, 2020. 1, 2
- [34] Hao Wu and Saurabh Prasad. Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification. *IEEE Transactions on Image Processing*, 27(3):1259–1270, 3 2018. 1
- [35] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11045 LNCS:3–11, 7 2018. 2
- [36] Zhi-Hua Zhou. Semi-Supervised Learning. In *Machine Learning*, pages 315–340. Springer Nature, Singapore, 2021. 1