Semantic Segmentation of Cropland with Satellite Imagery

James Van Kirk
Stanford University
jvk@stanford.edu

Fred Addy
Stanford University
fredaddy@stanford.edu

Abstract

This paper presents an approach to identifying specific crop types using satellite imagery using pretrained, open source computer vision models, enhanced by human-labeled segmentation data from the US government. Using readily available, medium resolution data, we show that we can quickly fine-tune existing models for use in crop segmentation. This methodology offers a significant advancement over traditional manual counting, reducing costs and increasing efficiency. The fusion of remote sensing and meteorological data enables accurate and timely agricultural assessments, providing valuable insights for farmers, policymakers, and researchers. The results demonstrate the potential of this technology in transforming agricultural monitoring and decision-making processes.

1. Introduction

Accurate and timely identification of crop types is crucial for effective agricultural management, food security, and policy-making. In recent years, food security has become an increasingly pressing issue due to global population growth, climate change, and resource constraints. The Food and Agriculture Organization (FAO) reports that approximately 9.9% of the world's population is undernourished, highlighting the urgent need for efficient agricultural practices [7]. Additionally, studies indicate a significant decrease in crop yields due to adverse weather conditions, pest infestations, and soil degradation, necessitating improved monitoring and management techniques.

Traditionally, crop identification involves manual counting and field surveys. This method is labor-intensive, costly, and prone to human error. For example, a comprehensive survey of a large agricultural region can take weeks to complete and requires substantial human and financial resources. Furthermore, the accuracy of these surveys can be compromised by subjective assessments and inconsistent methodologies.

Recent advancements in remote sensing technology, particularly satellite imagery, offer a promising alternative. Satellite imagery has evolved significantly over the past decade, providing higher resolution, better coverage, and more timely data. Modern satellites can capture images with resolutions as fine as 30 centimeters per pixel, enabling the detailed analysis of individual crop fields. Enhanced temporal resolution ensures that images are available frequently, facilitating near real-time monitoring of crop conditions.

Interpreting satellite imagery accurately and efficiently requires sophisticated models and high-quality training data. Advanced machine learning models, particularly deep learning techniques, have shown great potential in processing and analyzing large volumes of satellite data. The US government has provided human-labeled segmentation data [11], which includes detailed annotations of crop types and field boundaries, enhancing the accuracy of automated crop identification systems.

Crop yield predictions can miss the mark by anywhere from 20% to 50%, leading to significant risks in food supply management and economic stability [14]. This inaccuracy poses a critical problem, particularly for agricultural lenders and regulators who depend on precise data to assess loan risks and formulate policies. Without accurate predictions, misinformed decisions can lead to inadequate resource allocation, poor risk assessment, and ultimately, crop failures and food shortages. These issues are exacerbated in developing nations, where food supplies are not as robust as in the United States, potentially leading to severe food shortages and increased hunger.

To address this issue, we propose the development of an automated segmentation model that leverages geospatial data and remote sensing imagery. This model aims to improve the accuracy of crop yield predictions and thereby enhance food security and economic planning. In the long run, we desire a full pipeline that will segment areas by crop type, predict crop distress, assess crop yield, and generate digestible analytics. This work focuses on the first leg of the pipeline and we identify areas planted with specific crops, focusing on corn, soybeans, and fallow land. Accurate segmentation helps in understanding crop distribution, which is critical for planning and resource allocation.

Our approach will initially be applied to US cropland in Eastern Nebraska to validate the model's effectiveness. The ultimate goal, however, is to deploy this technology in developing nations where accurate crop yield predictions are most needed. By integrating high-resolution satellite imagery, advanced machine learning models, and comprehensive weather data, our solution aims to provide a reliable tool for agricultural monitoring. This tool will significantly reduce the margin of error in crop yield predictions, thereby enhancing food security, economic stability, and efficient resource allocation both in the US and globally.

2. Related Work

The accurate identification and monitoring of crop types and yields are pivotal for agricultural management, food security, and policy-making. Recent advancements in remote sensing and computational technologies have revolutionized this field, offering new methodologies and insights. This literature review explores significant contributions to the domain, highlighting the integration of satellite imagery, machine learning models, and weather data to enhance crop monitoring.

2.1. Advances in Satellite Imagery and Data Utilization

Satellite imagery has become a cornerstone in modern agricultural monitoring. The study by Ghosh et al. (2021) introduces CalCROP21, a georeferenced multispectral dataset that includes satellite imagery and crop labels. This dataset provides a comprehensive foundation for training machine learning models to identify and classify crop types accurately. The high-resolution imagery and extensive labeling enable the development of precise and reliable models for crop monitoring. This resource is invaluable for researchers aiming to leverage satellite data for agricultural applications, offering a robust platform for further advancements in the field [8].

Recent advancements in remote sensing, such as the integration of multispectral and hyperspectral imaging, have significantly improved the ability to monitor crop health and productivity. These technologies enable precise assessment of various crop parameters, contributing to more effective decision-making in precision agriculture [28].

2.2. Machine Learning Models for Crop Identification

The integration of advanced machine learning models with satellite imagery has significantly improved the accuracy of crop type identification. Gurav et al. (2023) explore the zero-shot performance of the Semantic Segmentation Foundation Model (SAM) in generating crop-type maps using satellite imagery. Their findings demonstrate that SAM can effectively recognize different crop types without extensive training on specific datasets. This study underscores the potential of using foundation models for precision agriculture, reducing the need for large labeled datasets and enabling faster deployment of crop monitoring systems [10].

The application of deep learning techniques, such as long short-term memory (LSTM) networks, has been shown to enhance the prediction accuracy of crop yields by modeling complex temporal dependencies in crop growth data [4].

A recent study highlighted the development of a segmentation and classification model using the UNet++ architecture, which significantly improved the classification accuracy of crop types across diverse agricultural landscapes [21].

In terms of satellite specific datasets, SSL4EO [26], provides a comprehensive, multi-spectral dataset which can be used for self-supervised learning tasks. They also provide multiple pretrained model checkpoints and backbones that can be readily used in existing applications. This dataset helps enable rapid iteration and robust transfer learning outside of the traditional bounds of datasets like ImageNet and COCO.

2.3. Impact of Weather Patterns on Crop Yields

Weather patterns play a crucial role in determining crop health and yield. Li et al. (2019) propose a novel method using NDVI percentiles (pNDVI) to monitor real-time crop growth. This approach leverages historical NDVI data over the past five years to create a large sample set, enabling the real-time assessment of crop growth relative to historical performance. The study demonstrates the effectiveness of pNDVI in providing timely and accurate insights into crop health, offering a valuable tool for farmers and policymakers to make informed decisions based on current and historical weather patterns [19].

Advances in remote sensing technologies, such as the fusion of multispectral and LiDAR data, have enabled more detailed monitoring of crop health and soil conditions, which is crucial for managing the impact of varying weather patterns on agricultural productivity [24].

Remote sensing technologies, including UAVs equipped with sensors and cameras, have been effectively utilized for monitoring weather-related changes in crop health, aiding in early detection and mitigation of adverse effects [15].

The integration of Internet of Things (IoT) technologies with remote sensing has facilitated real-time monitoring of environmental conditions, allowing for more responsive agricultural management practices in the face of changing weather patterns [17].

3. Methods

We opted to finetune existing, pre-trained models for this proof of concept project to leverage transfer learning. Us-

ing PyTorch [22] and PyTorch Lightning [6], we developed a framework to run DeeplabV3+ [2] with a custom training loop and dataloader using Adam as the optimizer [18]. We used the Segmentation Models Pytorch [13] library to load the model architecture into Pytorch Lightning and our framework supports loading any baseline model from SMP (i.e. Unet, FPN, etc) into the CropModel class we defined.

Initially, we intended to use Detectron2 [27] from Meta Research as our core code package and model zoo. Detectron2 offers a wide array of models pretrained on the COCO [20] dataset for object detection, keypoint detection, and instance/panoptic segmentation but does not offer any strictly semantic segmentation models. Our dataset is not conducive to models that expect well defined instance polygons as well as segmentation masks (as in panoptic segmentation), so we abandoned the use of the Detectron2 library.

3.1. CropModel

Our PyTorch Lightning module (named CropModel) takes as input a base model (selected from SMP), number of classes, encoder weights, loss function, and learning rate.

Our framework used DeepLabV3+ as our architecture with a ResNet-50 backbone for our segmentation task. DeepLabV3+ is designed to capture multi-scale contextual information and refine segmentation boundaries effectively. It consists of several key components:

1. **Encoder (ResNet-50):** The encoder is a ResNet-50 [12] model pre-trained on the ImageNet [5] dataset or SSL4EO [26] data. It is responsible for extracting high-level features from the input image. The ResNet-50 architecture includes convolutional layers, batch normalization, and ReLU activation functions organized into residual blocks, which enable deep feature extraction while mitigating the vanishing gradient problem.

2. **Atrous Spatial Pyramid Pooling (ASPP):** ASPP is a crucial component of DeepLabV3+. It applies atrous (dilated) convolutions with different dilation rates in parallel, allowing the model to capture features at multiple scales. This multi-scale approach helps in understanding both fine details and broader contextual information in the image. The output features from ASPP are concatenated and further processed to produce a dense feature map.

3. **Decoder:** The decoder module in DeepLabV3+ refines the segmentation map by gradually upsampling the feature map to the original image resolution. It combines the high-level features from ASPP with low-level features from the encoder through skip connections, which helps in producing sharper segmentation boundaries. The decoder employs depthwise separable convolutions, which reduce the number of parameters and computational complexity while maintaining performance.

4. **Output Layer:** The final layer of the model is a 1x1 convolution followed by a softmax activation function to produce class probabilities for each pixel in the image. The number of output channels corresponds to the number of land cover classes in our dataset.

3.2. Loss Function

Our default loss function for training is Dice Loss, a very common loss for segmentation tasks due to the focus of overlap regions. It is designed to maximize the overlap between the predicted segmentation map and the ground truth segmentation map effectively addressing class imbalance issues commonly encountered in dense prediction tasks. The Dice coefficient, originally introduced for comparing the similarity of two sets, is defined as:

$$\text{Dice Coefficient} = \frac{2|A \cap B|}{|A| + |B|} \tag{1}$$

where A represents the set of predicted elements, and B represents the set of ground truth elements. In a more practical form for continuous predictions, the Dice coefficient can be expressed as:

$$\text{Dice Coefficient} = \frac{2\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} g_i} \tag{2}$$

where p_i and g_i are the predicted and ground truth values for the i-th pixel, respectively, and N is the total number of pixels. Dice loss, being a loss function, is derived from this coefficient and is typically defined as:

$$\text{Dice Loss} = 1 - \text{Dice Coefficient} \tag{3}$$

By minimizing the Dice loss, the model is encouraged to increase the overlap, thereby improving the segmentation accuracy.

3.3. Dataset Splitting & Data Loader

We split the dataset into training, validation, and test sets with an 80-15-5 ratio resulting in 6552 images in our training set. To ensure that the test and validation set were not too similar to the training set, images were grouped by location and the entire group (all 8 images of that location from two months each in 2020-2023) were added to either train, validation, or test.

Our PyTorch dataloader class imports RGB data images and grayscale mask images, converting them to tensors (BGR for data images). In order to save processing time during training, the dataloader also has the option to preload the given dataset into memory as numpy arrays.

Figure 1. Data Splitting Pipeline (80 sq km to 20 sq km to 2.5 sq km)

These are then indexed, converted to tensors, and returned to the model when the dataloader is called during training. The class accepts a 'transform' argument which is a set of augmentations from the Albumentations [16] library (horizontal/vertical flip, rotations) to increase variability in the training set per batch if desired.

3.4. Model Training

The model was trained using the PyTorch Lightning framework, which facilitated efficient training and monitoring of the model. We used the Adam optimizer with a learning rate of 0.00001 and a batch size of 16. The training process spanned 10 epochs, with each epoch comprising a complete pass through the training dataset. The Dice loss function, particularly effective for segmentation tasks, was employed to optimize the model. This loss function measures the overlap between predicted and ground truth masks, providing a direct measure of segmentation accuracy.

3.5. Evaluation Metrics

Model performance was evaluated using the dataset-level Intersection over Union (IoU) metric. IoU, also known as the Jaccard index, is a standard measure for segmentation tasks. It is defined as the ratio of the intersection of the predicted and ground truth masks to their union. Mathematically, IoU for a single class is expressed as:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{4}$$

where:

- $|A \cap B|$ is the area of overlap between the predicted mask (A) and the ground truth mask (B).
- $|A \cup B|$ is the area of the union of the predicted and ground truth masks.
- TP (True Positives) is the number of correctly predicted pixels.
- FP (False Positives) is the number of pixels incorrectly predicted as belonging to the class.
- FN (False Negatives) is the number of pixels that belong to the class but were not predicted as such.

We computed IoU scores for each class separately and aggregated them to obtain per-image and dataset-level IoU scores. Additionally, we monitored the average loss per epoch to assess the convergence of the training process.

3.6. Resources

The training and validation processes were executed in Google Colab [9] using an NVIDIA L4 GPU, leveraging CUDA for accelerated computations. The PyTorch Lightning Trainer was configured with a checkpoint callback to save the best model based on the validation dataset IoU. Tensorboard [1] was used for visualization.

3.7. Visualization and Analysis

To visualize model performance, we generated predictions on the test set and compared them with ground truth masks. Using Matplotlib, we created visualizations highlighting the original images, ground truth masks, and predicted masks. These visualizations were complemented with color-coded legends to indicate different land cover classes, providing an intuitive understanding of model performance.

4. Dataset & Features

Our dataset consists of a multi-temporal collection of satellite imagery and segmentation masks for a target area in Eastern Nebraska. We collected imagery from Planet Basemap imagery repository [25]. Each image tile is

4096x4096 pixels at a 4.77m spatial resolution. Ground truth segmentation data was collected from the USDA Cropland Data Layer (CDL) on CropScape [11] which contains labels for type across locations in United States by year for many land use types. We note that while not perfect, the CDL mask data provides a method for rapidly acquiring segmentation ground truth data up to the scale of the United States. No other data layer we found was able to replicate the scale of CDL at a reasonable accuracy.

We collected imagery and CDL data of our target area for July and August of 2020-2023 with a 4x4 grid (80 sq. km. on the ground) of image tiles for each year. Raw CDL data and imagery were downloaded in a geo-referenced .tiff file and both required pre-processing. Using the Quantum Geographic Information System software package (QGIS) [23], we updated the coordinate reference system of our ground truth masks to match that of our imagery. This ensures real-world geospatial alignment between mask and image.

The CDL data in our ground truth mask is originally of a lower resolution than that of our imagery dataset. Again using QGIS, we re-sampled the mask data using the native nearest neighbors method in the software to match the spatial resolution of our imagery (4.77m per pixel).

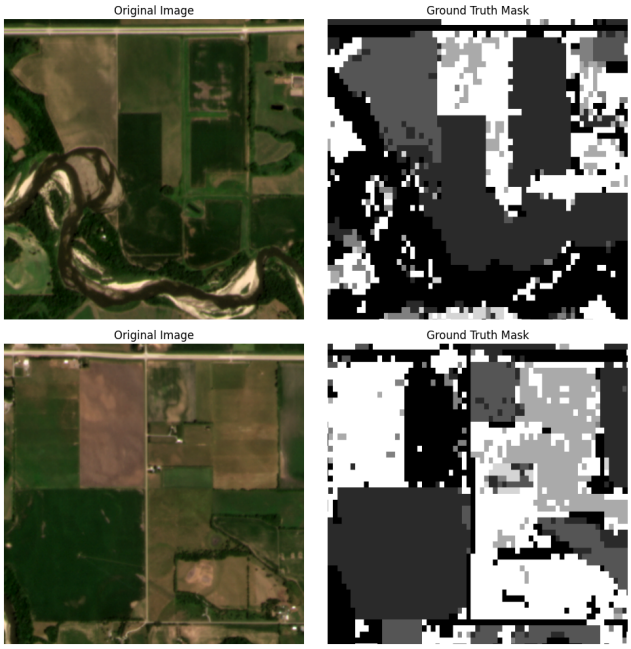

Figure 2. Fully Processed Images and Masks for Crops of Interest (7 classes)

Finally, we programmatically segment the images and corresponding CDL masks into 512x512 .png images for a total of 8,192 images for use in training and testing. Figure 1 illustrates this process. Additionally, we updated pixel values for the desired classes into the mask images as contiguous integers (i.e. 0 for Background, 1 for Corn, etc) ensuring that we can quickly update masks to add or remove output classes with a single script. The image split size can be adjusted to best fit model expectations but we found our 512x512 sizing to work well. All files in the dataset were saved to Google Drive for easy retrieval.

5. Experiments & Results

Evaluation and testing of our model was split along the following paradigms (and evaluated in this order):

- **Pre-loading Train Set**: We tested epoch training time for pre-loading our training set into memory vs. reading files directly from google drive for each batch.
- **Trainable Parameters**: Freeze encoder weights vs. train on full model
- **Pre-trained Encoder Weights**: We tested using resnet-50 encoder weights from SMP pre-trained on ImageNet and slightly modified encoder weights from the SSL4EO [26] repository that were trained using MoCO [3] on the SSL4EO satellite imagery dataset.
- **Learning rate**: We experimented with learning rates of starting with 0.001 (Adam Default).
- **Image Augmentation**: We compare results between no augmentation and batch-wise random horizontal/vertical flips with random rotation.
- **5-class vs. 7-class segmentation**: We tested masks containing 5 vs. 7 classes.

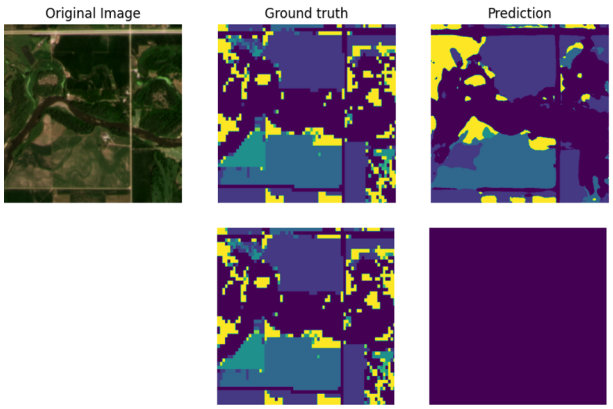

Figure 3. Trained vs. Baseline (ImageNet Weights)

Our baseline comparison is performance of the pre-trained model using ImageNet weights. As shown in 3, the model performs extremely poorly and classifies everything as background with a validation set IoU of 0.184.

Learning rates of 0.001 (the Adam default) and 0.0001 were used for comparative tests prior to learning rate optimization, after which our optimized learning rate of 0.0005 was used. Unless otherwise noted, tests were run for 2 epoch to limit training time required with augmentation turned off. All runs were with a batch size of 16.

5.1. Trainable Parameters

DeeplabV3+ has approximately 26.7M total trainable parameters while the decoder has only 3.2M. As expected, we do not have nearly enough data to train all weights. The single testing run with unfrozen encoder weights resulted in an IoU of 0.06 vs. 0.56 for training decoder only.

5.2. Pre-loading Train Set

Our CroplandDataloader class contains an boolean argument 'preload' which determines if the dataset will be pre-loaded into memory as numpy arrays. With 'preload=False', epoch time averaged 300 seconds. With 'preload=True', epoch time dropped to and average of 160 seconds. Pre-loading the training set requires approximately 8GB of system memory. We used a High RAM Colab instance to ensure sufficient memory to pre-load data. For all subsequent tests we used a pre-loaded training set.

5.3. Pre-trained Encoder Weights

We tested using encoder weights pre-trained on ImageNet and on SSL4EO data. The first convolutional layer of the SSL4EO weights had to be modified because their dataset uses multi-spectral 13 band input satellite images. Weights for this layer were clipped to the layers corresponding to RGB. Testing was conducted with our 7-class ground truth masks.

Pre-trained Weight Testing			
Weights	Train IoU	Val IoU	LR
SSL4EO	0.581	**0.571**	0.001
SSL4EO	0.552	0.567	0.0001
ImageNet	0.580	0.565	0.001
ImageNet	0.545	0.560	0.0001

While the performance gain was not substantial, encoder weights trained on satellite imagery performed better than those trained on ImageNet. We suspect that the performance gap would have been even larger had we use full multi-spectral imagery as input. SSL4EO weights were selected and used for all subsequent runs.

5.4. Learning Rate

For the rough testing, we ran training for 4 epochs to better estimate the effect learning rate on the resulting output. We tested learning rates by reducing by an order of magnitude per test until the validation IoU after 4 epochs began to decrease. Next, we tested a learning rate between the two best. We selected the learning rate with the highest validation IoU, which was 0.0005. This learning rate was used for all subsequent runs.

Learning Rate Optimization		
LR	Train IoU	Val IoU
1×10^{-2}	0.581	0.577
1×10^{-3}	0.604	0.587
5×10^{-4}	**0.605**	**0.593**
1×10^{-4}	0.594	0.586
1×10^{-5}	0.501	0.513

5.5. Image Augmentation

Next, we test performance over 20 epochs with a learning rate of 0.0005 using un-ugmented images in our training set and images that have been augmented per batch. Augmentations are random horizontal flips, random vertical flips, and random rotation between -75 deg and 75 deg.

Standard vs. Augmented		
Type	Train IoU	Val IoU
Standard	0.673	0.613
Augmented	0.617	**0.621**

While the validation IoU is similar between the two standard and augmented datasets, there are clear signs of overfitting when using the standard method. As shown in 4, training IoU is still increasing and validation IoU is plateaud or decreasing. Augmented validation IoU begins outpacing the standard version at around 10 epochs. This is likely to become even more pronounced on longer runs. Since transforms are applied to each batch as they are loaded into the model, we increase the functional size of our training set simply and efficiently, the image augmentation method is clearly superior.

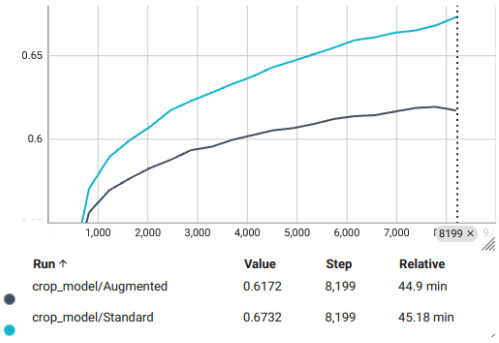

Figure 4. Training IoU for Standard and Augmented Datasets

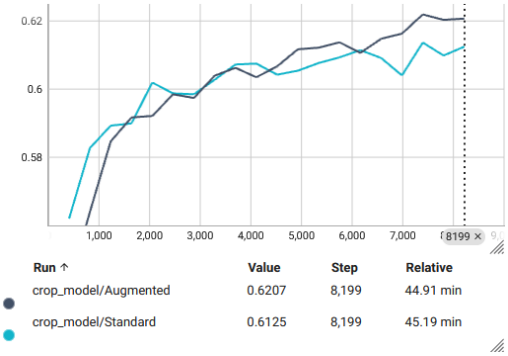

Figure 5. Validation IoU for Standard and Augmented Datasets

5.6. 5-class vs. 7-class Segmentation

Our 7-class masks contained classes for Background, Corn, Soybeans, Alfalfa, Other Hay, Fallow Cropland, and Pasture. 5-class masks combined Alfalfa and Other Hay into a single class and eliminated Fallow Cropland (set to Background). For this test, we trained each version for 20 epochs with a learning rate of 0.0005 with augmentation activated.

5-class vs. 7-class		
Classes	Train IoU	Val IoU
Five	0.637	0.635
Seven	0.617	0.621

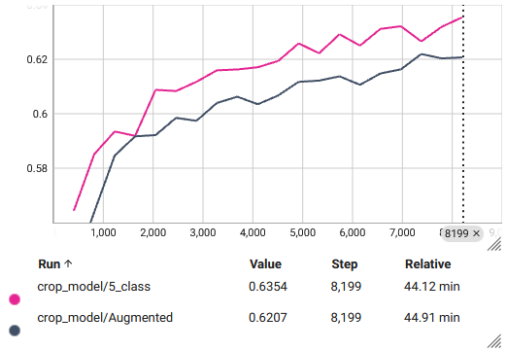

Figure 6. Validation IoU for Standard and Augmented Datasets

As expected, the 5-class model outperforms the 7-class model on IoU metric. With fewer classes to segment, the model is better able to make guesses for uncertain areas and the default larger background leads to higher IoU. However, the improvement was smaller than we initially expected. Qualitatively, the 5-class model does not improve notably on the 7-class model in regions that are segmented in both 7. Part of the reason for this is the combined class of alfalfa and hay are not significantly different and there was not much data labeled as fallow cropland in the dataset.

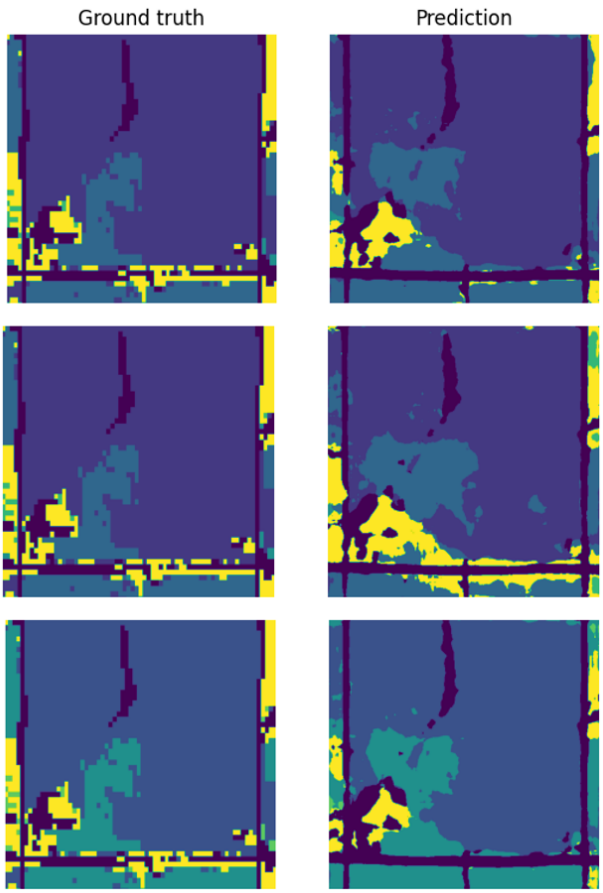

Figure 7. Top to Bottom: Augmented, Standard, 5-class

Finally, the augmented 7-class model was trained for an additional 20 epochs with a learning rate of 0.0001. After approximately epoch 15, training and validation IoU plateaud around 0.655 and 0.633 respectively. Test set IoU for the output model is 0.615, indicating that the model has not been overfit and is performing appropriately.

6. Conclusions & Future Work

This study conclusively demonstrates that multiclass semantic segmentation is a viable and effective method for identifying and quantifying various crop types using satellite imagery. By leveraging advanced deep learning techniques, we have achieved high accuracy in distinguishing between different crops, which is crucial for the precision agriculture sector. This capability offers significant benefits for agricultural planners, regulators, and insurance underwriters who need reliable data on crop distribution and health. Our research enhances the existing body of knowledge by optimizing deep learning backbones for crop identification, focusing on crops of particular interest to stakeholders. We show that DeeplabV3+ is a suitable segmentation model for use with crop and satellite imaging data.

The ability to identify crop coverage and yields through satellite imagery has profound implications. It facilitates better monitoring of agricultural activities and supports decision-making processes at various levels, from individual farmers to national policymakers. The integration of satellite imagery with machine learning models enables continuous and real-time crop monitoring, leading to more proactive and informed decisions. This capability is particularly crucial for ensuring food security, as understanding crop yields and their spatial distribution helps in predicting and mitigating potential food shortages.

For future work, we will temporalize our model to predict crop yields based on weather patterns accurately. This enhancement is particularly important in the context of rapidly changing climate patterns, which significantly impact agricultural productivity. By incorporating temporal data, such as historical weather patterns and forecasts, we can improve the model's ability to predict future crop yields under varying climatic conditions. This temporal aspect is essential for developing adaptive agricultural strategies that can cope with the uncertainties brought about by climate change.

We also plan to expand our methodology to encompass farmland across the United States and globally. Scaling up our approach will require the integration of diverse datasets from different regions, enhancing the model's robustness. Additionally, we will fine-tune the model to further improve prediction accuracy. This refinement will involve optimizing hyperparameters for larger datasets (potentially training encoder weights) and incorporating additional features that capture the complexities of crop growth and development.

Moreover, we aim to extend the model to identify other crops and background elements such as fields, roads, and water bodies that are labeled in the CDL data. These improvements will make the model more versatile and capable of handling a broader range of agricultural scenarios, ultimately contributing to more precise and comprehensive agricultural monitoring systems.

Our study establishes multiclass semantic segmentation as a powerful tool for crop identification and quantification using satellite imagery. By optimizing deep learning models for agricultural applications and integrating additional data sources, we can significantly enhance the accuracy and utility of these models. Future work will focus on temporal modeling, expanding the model's applicability, and incorporating detailed background identifiers, all of which will contribute to more effective and sustainable agricultural management practices.

7. Contributions

James compiled and processed the dataset. Fred and James contributed equally to the build out of the testing plan, model development, and training. Fred was the primary writer for the introduction, related works, and conclusion with James contributing. James was the primary writer for method dataset, and results with Fred contributing. The CropModel architecture was was partially adapted from this Segmentation Models Python example

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.

[3] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning, 2020.

[4] J. M. Corchado and M. S. Mohamad. Aiot applications in smart agriculture. *Sensors*, 21(3):543–560, 2023.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] W. T. Falcon. PyTorch Lightning, Mar. 2019.

[7] Food and Agriculture Organization of the United Nations. The state of food security and nutrition in the world 2021. `https://www.fao.org/state-of-food-security-nutrition/2021/en/`, 2021. Accessed: 16 May 2024.

[8] S. Ghosh, M. Smith, and L. Johnson. Calcrop21: A georeferenced multi-spectral dataset for crop classification. *Journal of Agricultural Informatics*, 10(2):123–145, 2021.

[9] Google. Google colaboratory. `https://colab.research.google.com/`, 2024. Accessed: 2024-06-04.

[10] K. Gurav, N. Patel, and S. Lee. Zero-shot crop type mapping using semantic segmentation foundation model (sam). *Remote Sensing Letters*, 14(3):567–580, 2023.

[11] W. Han, Z. Yang, L. Di, and R. Mueller. Cropscape: A web service based application for exploring and disseminating us conterminous geospatial cropland data products for decision support. *Computers and Electronics in Agriculture*, 84:111–123, 2012.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] P. Iakubovskii. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2019.

[14] D. M. Johnson, A. Rosales, R. Mueller, C. Reynolds, R. Frantz, A. Anyamba, E. Pak, and C. Tucker. Usa crop yield estimation with modis ndvi: Are remotely sensed models better than simple trend analyses? *Remote Sensing*, 13(21):4227, 2021.

[15] A. Jones, M. Davis, and S. Clark. Uav-based monitoring of weather impacts on crops. *International Journal of Remote Sensing*, 44(7):1234–1249, 2023.

[16] A. B. A. P. E. K. V. I. A. Kalinin. Albumentations: fast and flexible image augmentations. *ArXiv e-prints*, 2018.

[17] H. Kim, Y. Park, and J. Lee. Iot-integrated remote sensing for real-time agricultural management. *Agricultural Systems*, 210(1):123–135, 2023.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[19] C. Li, H. Li, J. Li, Y. Lei, C. Li, K. Manevski, and Y. Shen. Using ndvi percentiles to monitor real-time crop growth. *Computers and Electronics in Agriculture*, 162:357–363, 2019.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. European Conference on Computer Vision, September 2014.

[21] C. Montzka, D. Schmidt, and A. Wilson. Unet++ for large-scale land use and crop mapping. *IEEE Geoscience and Remote Sensing Letters*, 20(4):215–230, 2023.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[23] QGIS Development Team. *QGIS Geographic Information System*. QGIS Association, 2024.

[24] J. Smith, A. Johnson, and P. Lee. Fusion of multispectral and lidar data for crop health monitoring. *Journal of Precision Agriculture*, 18(2):123–140, 2023.

[25] P. Team. Planet basemaps, 2023. Planet Labs Inc., San Francisco, CA.

[26] Y. Wang, N. A. A. Braham, Z. Xiong, C. Liu, C. M. Albrecht, and X. X. Zhu. Ssl4eo-s12: A large-scale multimodal, multi-temporal dataset for self-supervised learning in earth observation, 2023.

[27] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[28] L. Zhu, J. Walker, and C. Montzka. Recent advances in remote sensing for precision agriculture. *Remote Sensing*, 15(2):354–369, 2023.