Crop Anomaly Detection with Semantic Segmentation

Mark Lisi

December 2023

1 Introduction

Agriculture is the most foundational component of human civilization, and is one that we are still heavily reliant on across the world. As climate change becomes an increasingly unavoidable facet of everyday life, it stands to reason that we should apply the strongest tools of our generation to monitoring and preserving the well-being of our crops. With modern drone technology aerial photographs of farmland are easily collectible, making the field of computer vision a natural fit for diagnosing issues in potential harvests before it becomes too late to save them.

In this project we use an implementation of UNet, a popular convolutional neural network architecture, to carry out semantic segmentation on a large dataset of exactly such aerial crop images. Our areas of interest are different types of agricultural anomalies - we focus primarily on drydown, x, and y.

Perhaps the greatest challenge that this project posed was wrestling with a dataset that is impractically large for one student. The systems used in these experiments have limited RAM/VRAM, and as such were not able to read in the entire dataset at once. Despite this, using efficient algorithms for dynamically loading in files and operating on a subset of the given data we were still able to achieve satisfactory results that have legitimate real-world applications with a human in the loop.

2 Data

For this project we used the 2021 Agriculture-Vision dataset. The dataset is comprised of aerial photos of farmland with a very high level of detail (resolutions as high as 10cm per pixel). Each image contains RGB channels and a near-infrared (NIR) channel. This NIR channel may prove useful for understanding plant health, since a healthier plant with more chlorophyll will reflect more near-infrared energy than an unhealthy plant. The data also contains binary masks highlighting the affected area in each image to act as labels. This structuring introduced the first difficulty of processing this data: each of the

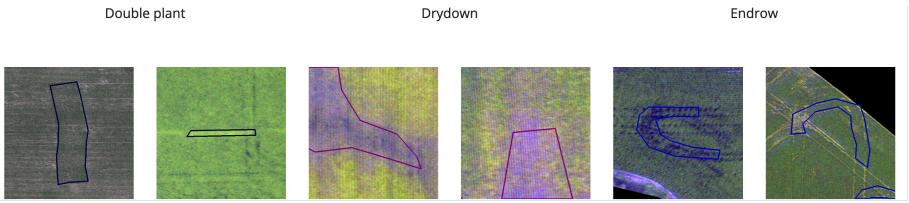


Figure 1: Examples of image data with ground-truth segmentation

nine anomaly classes has its own folder of labels. In each of these folders there are tens of thousands of 512x512 images - most of which are blank. The labels in each folder are aligned with corresponding RGB/NIR farmland images; there is only one type of anomaly present in each image, so 8 of 9 labels in each folder will be blank at any given position. Strategies for managing this obstacle/data preprocessing steps are detailed in the following section.

3 Methodology

Before we begin the computer vision/machine learning components of this project, we must handle and preprocess the dataset. Due to aforementioned hardware constraints, we cannot effectively load in the entire dataset at once - instead, we can extract all labels of a given class with no extraneous blanks and train a model on each class individually. We do this by leveraging file naming conventions in the dataset and picking out the labels with a maximum value of 1 rather than 0.

Once the labels are handled, we then stack the 3 RGB channels with the additional NIR channel to fortify our features and give the model more information to learn from.

To extract the positions of anomalies from these images we chose to use UNet, a popular neural network architecture that uses an encoder to capture contextual information and a corresponding decoder with skip connections to recover spatial detail. For our particular implementation of UNet we include two convolutional layers with a 3x3 kernel in both the encoder and the decoder. The encoder finishes with a max pooling layer with a 2x2 kernel size and a stride of 2, while the decoder has a bilinear upsampling layer. We choose to use convolutional layers with 64 neurons to mitigate computational expense. Each convolutional layer is passed through the ReLU activation function to introduce nonlinearity.

Our loss function is cross-entropy loss, a commonly used objective function for classification tasks in machine learning. In a general case the function can be formulated as

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

where y_i represents the true probability distribution, $\hat{y}_i$ is the predicted probability distribution, and C = the number of classes. In our case we will only be training on two classes at a time (0, or "background", and 1, the given class of anomaly). We can simplify the objective function in this case to

$$L(y, \hat{y}) = -\left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)\right)$$

4 Implementation Details

We use the aforementioned UNet architecture (exact implementation in attached code). To minimize the already hours-long runtimes of training this model we use a smaller batch size of 16. We use the relatively standard Adam optimizer with a learning rate of 0.001.

To efficiently preprocess the data we define a custom PyTorch `Dataset` and overload the `__getitem__` method. The dataset stores accepts directories for images and labels as well as a list of indices corresponding to the positions of non-zero labels for a given class. These indices are identified by using the default file naming conventions of the dataset to programmatically read only labels with information in them - each filename has an alphanumeric prefix corresponding to a plot of land, but many photos often come from one plot. If the first element with a certain prefix is an empty label, we know all with that prefix will be! Our custom `Dataset` object stores filepaths for all non-zero labels and their corresponding RGB/NIR channels, only loading them in when needed through the `Dataloader`. Our overloaded `__getitem__` method automatically reads the necessary files, concatenates the RGB/NIR channels, and returns an image tensor and a label tensor for the model.

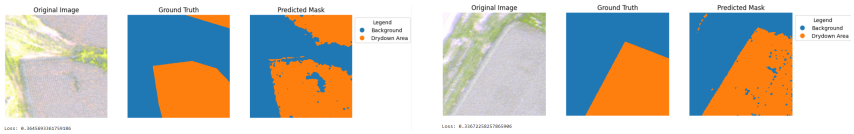
The first iteration of the model was trained over 5 epochs, but saw no improvement in loss after the first epoch - in the interest of runtime, following iterations were trained over a single epoch.

5 Results

Since a practical use case for this model is qualitative inspection by a farmer or other user, our measure of success is somewhat qualitative as well. The ideal outcome would be mapping each ground truth mask exactly, and from basic human observation of our results it seems that cross-entropy loss does not always correlate strongly with that metric; some very well matched masks have higher loss than poorly matched ones. Many of the predicted masks exhibit "graininess", where an otherwise contiguous prediction has small bits omitted in the middle - with a human farmer in the loop, these are easily distinguishable as a quirk in the model.

The greatest weakness of the model is its tendency to predict a blank mask when confused. This happens most frequently on poor quality images, and training the model for more epochs could likely mitigate the issue - still, the

model demonstrates impressive predictive power on images that are essentially unintelligible to the human eye. The model also tends to predict a blank mask when the anomalous area in the image in question is very small. This issue might be trickier to completely remove, but implementing a weighted loss function that discourages the prediction of the background class is one approach worth exploring.



(a) (b)

Figure 2: Strong matches for drydown generated by the model

(a) (b)

Figure 3: Robust identification of overwatered area in poor-quality images

(a) (b)

Figure 4: Grainy predictions with relatively high loss that are still useful

In general, a low loss did not always seem to represent a great prediction and vice versa: the average loss of our models improved after training, but not by a significant amount. This incongruity between loss and performance at the actual task suggests that a different objective function could improve results. A commonly used loss metric in semantic segmentation tasks is mean intersection-over-union (mIoU) loss - implementing a custom mIoU loss function may have improved the predictive power of the model.

6 Conclusion / Future Work

Despite a host of technical limitations, this project still makes clear that deep learning techniques in computer vision have directly useful applications in the fields of agriculture and environmental science.

As is evident throughout this report, the greatest issue posed in this project was the size and structure of the dataset. Future work could be carried out by moving these experiments to Yale's HPC clusters and training a deeper model on the entire dataset. A few specific techniques that could likely improve performance with greater compute are:

- increasing the number of neurons in each convolutional layer
- adding more convolutional layers
- training over multiple classes at once
- training over a larger chunk of the dataset, if not the entire thing
- using a larger batch size or lower learning rate

Still, many of the results from our experiments could undoubtedly prove useful to a farmer trying to diagnose their ailing crops!