# Semantic Segmentation in Agriculture

**Raunaq Rewari**
Department of Management Science and Engineering
Stanford University
raunaq@stanford.edu

## Abstract

Precision agriculture in its most recent form leverages the state of the art in Artificial Intelligence to selectively apply seeds, fertilizer and other inputs to a field. This project was done in collaboration with Blue River Technology, a company that is trying to reduce the amount of herbicide use in the field by identifying crops and weeds. This project explores the semantic segmentation approach in identifying crops and weeds in Soybeans fields. This dataset is extremely imbalanced especially with regards to weeds since the images were taken in fields with low weed pressure (less number of weeds in the field). The U-net (Olaf et al. [2015]) architecture was modified to include residual blocks (He et al. [2015]) instead of the usual convolution layers. Another major contribution of the paper was evaluating different loss functions, which is especially important for a heavily unbalanced dataset like this. The weighted multi-class soft dice loss performed the best with the model doing pretty well in segmenting crops from background with a precision of 64% and recall of 99%, but understandably couldn't really do a very good job at segmenting weeds from background giving a mean Intersection over Union (mIoU) score of 0.35.

## 1 Introduction

This project's data was sponsored by Blue River Technology, which is a precision agriculture company that is trying to fundamentally change how farming is done. Their first product, "See-and-spray" is designed to reduce the amount of herbicide usage on a typical farm by 90% by using the latest advancement in Computer Vision to identify crops and weeds in a farm and selectively spraying herbicide only on the weeds. This is in stark contrast to the prevailing practice of broadcast spraying over the entire area of the farm which not only increases the amount of money a farmer spends in a particular growing season but also makes weeds herbicide resistant. This requires that new herbicides come to the market each year and these are usually more expensive than the previous iteration and also necessitates in some cases the use of GMO seeds which are significantly more expensive than regular seeds. As is apparent, this has the effect of slowly increasing the cost of inputs each year, causing many farmers to feel the pressure of farming. This increased cost gets passes down to consumers which is unsustainable in the long run, considering the population growth. This project will help move towards the goal of improving the food security across the globe.

The current methodology to solving this problem of identifying crops and weeds at Blue River Technology is a bounding box approach, where the output of the model is bounding boxes with the highest confidence for both crops and weeds. This has worked very successfully for cotton where the crops are fairly spaced out. Soybeans on the other hand have a more dense canopy that outputs a lot of bounding boxes that makes it hard to have a high resolution on spraying. As you can imagine, the higher the resolution of the detections, the higher the resolution

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Figure 1: Figure showing the input image to the model on the left and the corresponding annotations on the right. (Green = Crop; Red = Weed)

of the herbicide spraying and hence, lower the amount of herbicide used. The following figure (1) shows a typical example. Hence, for Soybeans, there was a need to move to semantic segmentation that could give pixel-wise classification for each image to be able to spray more accurately.

The input to the model is the RGB image on the left in 1. The labels for each input image is single channel image with the same dimensions as the input image and each pixel labeled as one of [0,1,2] corresponding to ground, crop, weed. A U-net He et al. [2015] derived model is trained to predict the class for each pixel of the image. Like is common for most semantic segmentation papers, the mean Intersection over Union is reported.

## 2 Related work

To the best of my knowledge, there is no existing work on image segmentation on real images of fields. However, there is a lot of research done on semantic segmentation by the larger computer vision community.

The general idea followed by the most successful recent architectures is that of an encoder-decoder strategy, where the encoder is responsible for condensing the most important information and the decoder is responsible for bringing back that information to the appropriate size/context. The original paper by Jonathan et al. [2014] lead to an invigorated interest in using deep learning for semantic segmentation. They introduced the concept of fully convolutional networks to allow images of all sizes and inputs and also introduces the concept of learnable up-conv/deconv layers to increase the size of the output of the model to the image size. They used skip connections to layers of different resolutions to capture global context.
The U-net architecture proposed by Olaf et al. [2015] has a contracting path of convolution and pooling layers to capture context and a symmetric expanding path that enables precise localization. This model architecture is good in the sense that it enables good localization by virtue of copying over the layers from the contracting path and also uses larger context, which is crucial for segmentation tasks.
The SegNet architecture proposed by Vijay et al. [2015] is another encoder-decoder based model architecture that uses a different approach to upsampling, instead of copying over layer like in U-net. The novelty of SegNet lies is in the manner in which the decoder upsamples its lower resolution input feature maps. Specifically, the decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to upsample. The upsampled maps are sparse and are then convolved with trainable filters to produce dense feature maps.

The Fully-convolutional denseNets for semantic segmentation by Simon et al. [2016] uses the idea of DenseNets (Gao et al. [2016]) applied to the semantic segmentation task.
The DeepLabv3+ paper by Liang-Chieh et al. [2018] uses both the Spatial pyramid pooling (SPP) module and the encode-decoder structure we have been talking about. The SPP module able to encode multi-scale contextual information by pooling features at multiple rates and with multiple fields-of-view. The encoder-decoder structure as we have seen is used to capture sharper object boundaries by gradually recovering the spatial information.

In conclusion, it seems that most papers use the encoder-decoder strategy while changing the way the global information is captured.

## 3 Dataset and Features

The dataset for this project was sponsored by Blue River Technology. The data was collected using carts that were pushed through rows of crops in Soybeans fields in the US. There were 7379 images in total, 6700 of which were used for training, 379 for validation and 300 for testing. Since the images are pretty homogeneous in general, the images were randomly shuffled and split up into train/val/test.

Each input image was a RGB image with a resolution of 384x384. Since accurate labeling of the images is crucial for segmentation, the crowdsourcing service Figure8 Was used to annotate these images by the inhouse agronomists at Blue River Technology. This ensured that we were learning the right features, since its hard for a layman to differentiate between crop and weed. Like was mentioned earlier, the label corresponding to each image was a another single channel 384x384 image with each pixel labeled as one of the 3 classes (background, weed, crop). A sample has already been shown in 1

To make the training robust, a few augmentation techniques were used:

- Random Flips: The data was randomly filled in all directions
- Blur: Blurring was added to prevent over reliance on crisp looking images since some images in the dataset can be out of focus
- Color temperature: The color temperature of the images was also modified across a range to prevent over reliance on having collected good quality images. Some images had light leaks which we wanted to be robust to.

After looking at the predictions from the initial few trained models, it could be seen that the model was doing a decent job at picking the crops and their shape, but was throwing off speckled predictions for the weeds i.e. the predictions did not form a continuous area, but rather sparse spots. This prompted a more detailed look at the dataset, especially the labels. It was found that the labelers had labeled weeds even if they were 2-5 pixels wide and these kind of labelings seemed to be in huge number in the dataset. Since such minute weeds were even in the practical setting not a threat to the farm, OpenCV was used to clean up these regions for both crops and weeds to encourage the model to learn cleaner representations.

## 4 Methods

The U-net model, originally developed by Olaf et al. [2015] was used to train on this dataset. The architecture was modified to use residual blocks 2 instead of vanilla convolution layers in the original U-net implementation. The residual blocks help in attaining a deeper network without the downside of having a vanishing gradient problem.
Each convolution layer, except the last output layer in the model was with a 3x3 kernel and "same" padding to ensure same size of output. The last convolution to get the logits was with a 1x1 kernel and 3 channels corresponding to the 3 classes that were being predicted. Each convolution layer, except the last was also followed by a batchnorm layer and a relu activation layer. The model architecture can be seen in 3 Since there was a fundamental class imbalance issue, choosing the loss function was extremely important. Here are a few that were tried:

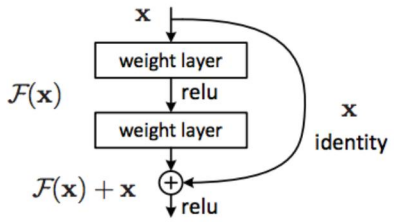


Figure 2: Figure showing a residual block

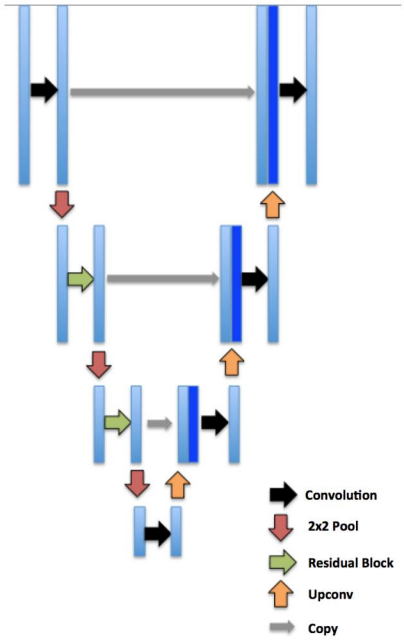


Figure 3: Figure showing the model architecture used

- Cross-entropy Loss: The regular cross-entropy loss was the first option tried and as expected did not perform well.
- Weighted cross-entropy Loss: This was done to ensure that the labels that were underrepresented were able to have sufficient gradient backpropogated to them. 2 weighing schemes were tried:
  - Class weights decided by hyper-parameter search
  - Class weights calculated by the inverse frequency of labels in a mini-batch
- Wasserstein Dice Loss: This is a dice loss in case of multiple classes as first shown by Fidon et al. [2017] in training CNN models.
- Weighted Multi-class Dice Loss: This was calculated by calculating the Dice score for soft binary segmentation for each class and adding it up, as shown in 4. Dice score is basically a measure of intersection over union and using this loss function means that we are trying to make sure that the predictions have maximum overlap with the ground truth while also ensuring good precision. Individual Dice score for soft binary segmentation where g is the ground truth and p is the predictions, is calculated as follows:

$$\text{D(p,g)} = \frac{\sum_i 2.g^i.p^i}{\sum_i g^i + p^i}$$

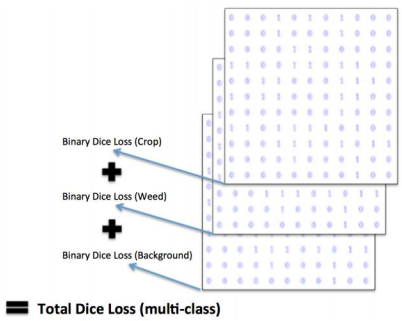


Figure 4: Figure showing calculation of multi-class dice loss

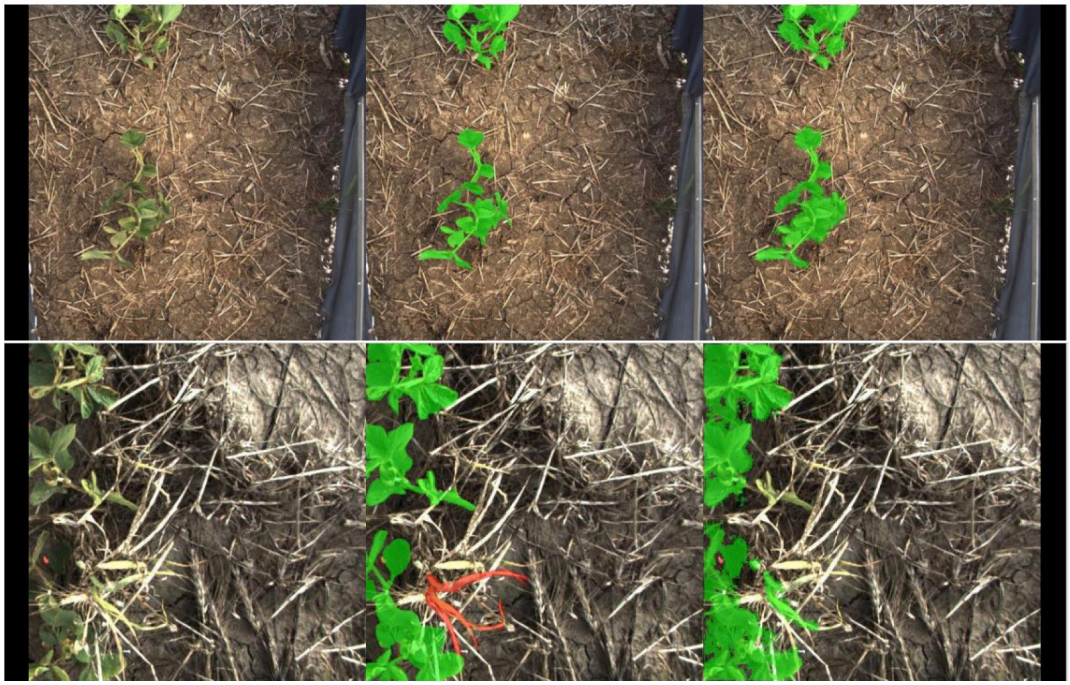

Figure 5: Figure showing results on two images, From left to right: Input image, ground truth, predictions

## 5 Experiments/Results/Discussion

The two sets of images shown in 5 are 2 examples from the dataset along with the predictions from the best performing model - model with the weighted multi-class dice loss. As can be seen from the first panel, the model does really well in segmenting crop, since it is relatively well represented in the dataset. However, looking at the second panel of images, it can be observed that the model doesn't do as good a job on weeds. This causes the mIOU (mean Intersection over Union) of the models to be low.
There could be a few reasons for why the model is not doing well on weeds:

- There are a very low number of images with weeds in them. For the images that do have weeds, they occupy a very small part of the image as shown in panel 2 of 5.
- The size of the images (384x384) is too small. There are not enough features to be learnt from the weeds.
- The weeds look a lot like the background in terms of shape in some cases (Panel 2 of 5) or look similar to crops in color preventing the model from learning.

The results are reported for the best performing models after doing a hyper-parameter search on learning rate and learning rate schedule. An interesting learning rate schedule that we are calling the "triangular" learning rate was tried, where the learning rate was gradually increased from the min learning rate of 0.0001 to max of 0.01 many times over the duration of training to prevent being stuck in saddle points. A batch size of 16 was used; any lower would not give a good representation of labels in the mini-batch and any higher would not fit in memory.

Here the mean intersection over union over all 3 classes for the test data is reported. Most papers on semantic segmentation report this metric.

| Loss-type | Test mIOU |
|---|---|
| Weighted cross-entropy | 0.13 |
| Wasserstein | 0.13 |
| Weighted Multi-class Dice Loss | 0.35 |

## 6 Conclusion/Future Work

The best performing model was achieved with the weighted multi-class dice loss model with a precision of 64% and recall of 99% for crops.

Given more time, the following directions will be explored:

- Try other models like Segnet which do not rely on learning the upsampling weights
- Try augmenting the dataset by training a GAN to generate more weeds
- Try running the model on higher resolution images to get more features to work with for weeds
- Prioritize image collection from high weed density fields

## 7 Contributions

I, Raunaq Rewari, was the sole person working on this project and this project was submitted as part of this class (CS230) only.

## References


Lucas Fidon, Wenqi Li, Luis C. Garcia-Peraza-Herrera, Jinendra Ekanayake, Neil Kitchen, Sebastien Ourselin, and Tom Vercauteren. Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. *eprint arXiv:1707.00478*, 2017.

Huang Gao, Liu Zhuang, van der Maaten Laurens, and Weinberger Kilian Q. Densely connected convolutional networks. *eprint arXiv:1608.06993*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Long Jonathan, Shelhamer Evan, and Darrell Trevor. Fully convolutional networks for semantic segmentation. *eprint arXiv:1411.4038*, 2014.

Chen Liang-Chieh, Zhu Yukun, Papandreou George, Schroff Florian, and Adam Hartwig. Encoder-decoder with atrous separable convolution for semantic image segmentation. *eprint arXiv:1802.02611*, 2018.

Ronneberger Olaf, Fischer Philipp, and Brox Thomas. U-net: Convolutional networks for biomedical image segmentation. *eprint arXiv:1505.04597*, 2015.

Jégou Simon, Drozdzal Michal, Vazquez David, Romero Adriana, and Bengio Yoshua. Encoder-decoder with atrous separable convolution for semantic image segmentation. *eprint arXiv:1611.09326*, 2016.

Badrinarayanan Vijay, Kendall Alex, and Cipolla Roberto. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *eprint arXiv:1511.00561*, 2015.