Tackling Class Imbalance on Agriculture-Vision Semantic Segmentation

G012 (s2118610, s2091900, s2122286)

Abstract

We focus on improving DeepLabV3+'s performance on the Agriculture-Vision semantic segmentation task by addressing the class imbalance present in the data. This is done by experimenting with different loss functions and augmentation and oversampling schemes. We manage to achieve competitive results with an mIoU of 46.40% on the test set by adding an Adaptive Class Weighting mechanism on the loss function of the Baseline Model.

1. Introduction

Research in pattern recognition and computer vision was catalysed by the introduction of ImageNet (Deng et al., 2010), a large-scale data set for image classification. Algorithms based on Deep Neural Networks are widely used, as they have proven to be efficient in several different tasks and datasets across numerous domains.

An important application field of visual recognition algorithms is agriculture. A promising avenue in this domain is segmentation of aerial images with deep learning techniques, as it might help prevent losses and increasing potential yield (Kamilaris & Prenafeta-Boldú, 2018). However, despite its importance, this area has seen relatively slow progress. To tackle this problem, Chiu et al. (Chiu et al., 2020b) proposed a large-scale farmland image dataset for semantic segmentation of agriculture patterns, Agriculture-Vision. Semantic segmentation consists in dividing an image into different segments and then marking all objects of the same type with the same label. In agriculture, it is necessary to process aerial farmland images, which means that semantic segmentation requires inference over remarkably large images with severe sparsity of the annotations, and highly imbalanced classes in the dataset.

In this study we tackle the class-imbalance problem. We use as a baseline the DeepLabv3+ (Chen et al., 2018b) semantic segmentation architecture. To tackle the class imbalance, we couple different loss functions to the architecture, perform data augmentation and oversampling with the hopes of increasing the performance of DeepLabv3+ on the Agriculture-Vision dataset by improving its performance on the sparse and imbalanced classes. This study thus contributes with an extensive exploration into multiple ways in which class imbalance can be addressed on Agriculture aerial imagery.

In the interim report, we planned on proposing a novel architecture that combined DeepLabv3+ and MSCG-Net (Liu et al., 2020) to address both the class imbalance and sparsity problems. However, due to lack of time to conduct our experiments, we decided to focus on the class imbalance problem using the DeepLabv3+ architecture as the baseline.

2. Data set and task

We will perform a semantic segmentation task on the Agriculture-Vision data set (Chiu et al., 2020b), which consists of high-quality aerial farmland images with labelled agricultural patterns. More specifically, we will use the challenge data set[1] used for a public competition ran in 2020 (Chiu et al., 2020a), which is a subset of the full data set. This subset contains 21,061 images of 512×512 pixels in four channels: RGB and Near Infra-red (NIR). We follow the split used in the competition and use 12,901 images for training, 4,431 for validation, and 3,729 for testing. These 512×512 are image patches from large farmland images that were cropped around annotated regions in the image.

Figure 8 contains a sample image of each class overlapped with the corresponding label for the RGB channels. As shown in the figure, each class has a corresponding color mask which indicates where the pattern represented by each class is located. From these examples we can observe the nature of the classes: Cloud shadow is normally homogeneous across the image, with big patches covering the image. Double plant and Planter skip are normally thin and don't cover a big portion of the image, which results in sparse classes (more on this later). Standing water is normally a big organic patch on the image, Water way is normally a geometric big stripe on the image, and Weed cluster represents fairly big organic clusters in the image.

The segmentation task consists in classifying each of the 262,144 pixels from an image into one of the aforementioned six classes (or background) representing an agricultural pattern in the farmland. To evaluate our models, we follow the framework established by (Chiu et al., 2020b) and also followed in the challenge (Chiu et al., 2020a). Hence, we use as evaluation metric the mean Intersection-over-Union (mIoU) given by:

$$mIoU = \frac{1}{c} \sum_{c} \frac{A(P_c \cap T_c)}{A(P_c \cup T_c)}$$

Where c denotes the number of classes (7, in our case, for the 6 patterns and the background); $A(P_c \cap T_c))$ is the area of overlap between the predicted mask for class c, P_c, and the ground truth mask for that same class, T_c; $A(P_c \cup T_c)$

[1] https://www.agriculture-vision.com/dataset

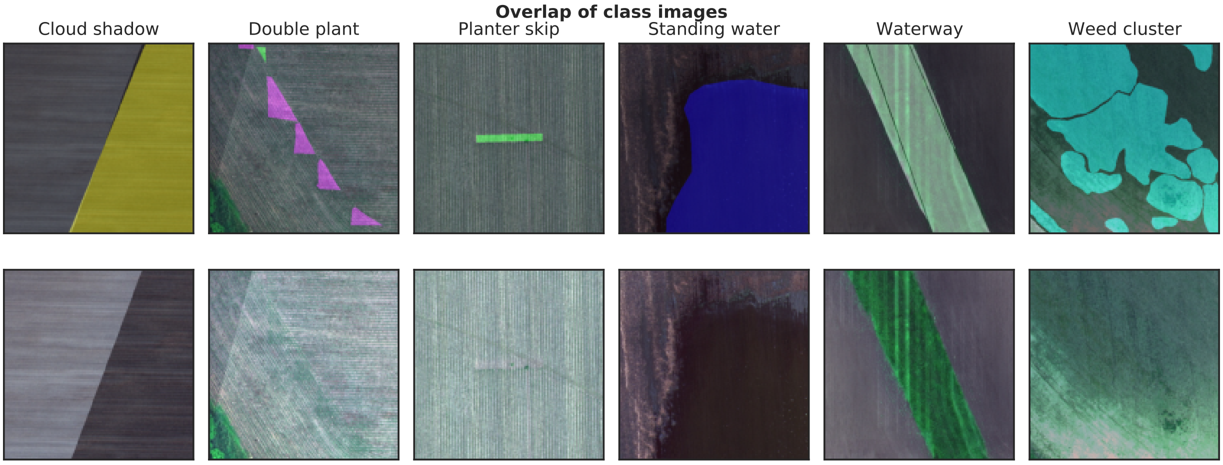

Figure 1. Sample images with overlay of labels for each class (Top), and sample images for each class without overlay (Bottom). We didn't include the first class, background, since it is present in all images. Each labeled pixel was assigned a different color to better visualise the label and prediction, yellow for Cloud shadow, magenta for Double plant, green for Planter skip, cian for Standing water, white for Waterway, blue for Weed cluster and black for the background.

denotes the area of the union of both the predicted and the truth masks.

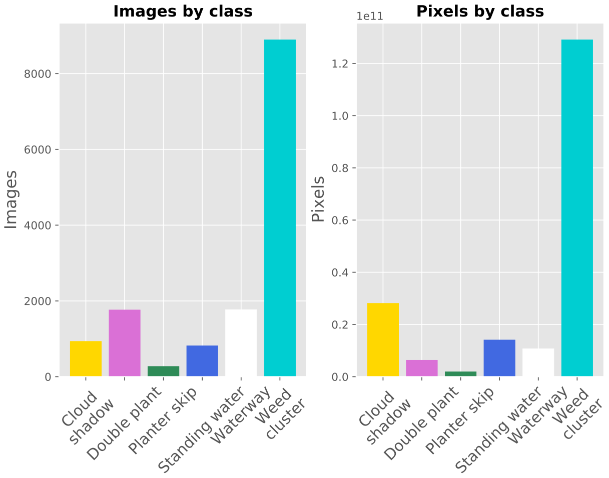

Figure 2. Number of images and pixels annotated with each class. We can see that the *Double plant* and *Planter skip* classes are highly underrepresented pixel-wise, compared to the rest of the classes.

However, in this data set it is possible to have overlap of different classes. To account for this, an adjustment is made to the standard mIoU metric. For pixels which have multiple labels, each class prediction is considered individually. This means that a prediction of any of the correct classes of the pixel will be considered as part of the intersection with the truth mask of that class $P_c \cap T_c$. Naturally, if the prediction does not include any of the ground truth classes, it will be considered incorrect in all of them.

Another important aspect of this data set is that it has a class imbalance. Almost 70% of the images have an annotation of the most common class, Weed Cluster, whereas, there are 4 classes with an annotation on less than 5% of the images. The number of images by annotation class is shown in Figure 2. In addition to this, different classes have different sizes and shapes and appear at different frequencies. This is reflected on the total number of pixels for each class, with Weed cluster having significantly more pixels than all the other classes. Indeed, for this task, the fact that we have more annotated images with a certain class than another, doesn't mean that this class is more represented. For this we have to look at the number of training pixels of that given class. For example, Double plant has more annotated images than Cloud shadow, but much less pixels. This means that Double plant is sparse.

To deepen this analysis we look at the percentage of pixels annotated with a given class in an image where that class is present. We call this measure the pixel coverage of the class on that image. We analyse the pixel coverage of each class in all images (where the class is present, which results in Figure 3. This allows us to further understand the sparsity of each class. For instance, we can see that Cloud shadow, due to its homogeneous shape has a broad spectrum of coverage over the images. Weed cluster, Waterway and Standing water also count with a less sparse pixel distribution. On the contrary, we can conclude that Double plant and Planter skip have a high sparsity and are less represented in the data set. We can see that there are a lot of images from these classes that have a very small coverage, i.e. too little pixels labeled with that class. These problems needs to be addressed to achieve a strong performance.

In principle, no further pre-processing is required beyond that already done by Chiu et al. (2020b). However, we performed some data augmentation, which will be explained in detail in Section 3.3, to combat the class imbalance problem mentioned above.

3. Methodology

In this section we go over the methods used by providing a detailed description into the architecture used for the given

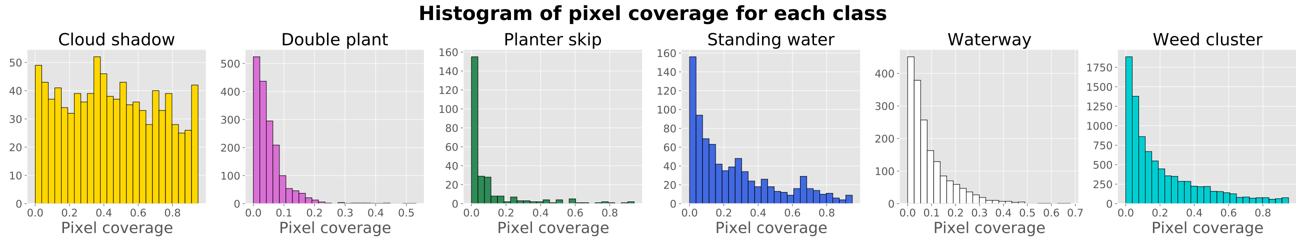

Figure 3. Histograms of proportion of pixel coverage for each class. Dense classes have a wide range of pixel coverage, whereas the sparse classes normally have a lot of images that have a very small coverage. This sparsity hinders the ability of the classifier to learn how to make predictions precisely, since there's too little pixels to learn from.

task, and the methods implemented to address the class imbalance problem.

3.1. Architectures & Related work

This section goes through the basic architectures for semantic segmentation, and introduces the baseline architecture that we are going to use in our study to address the class imbalance problem in the data.

3.1.1. The basic Semantic Segmentation architecture

The previously described dataset and task require a model that is able to identify irregular shapes, of any size, in a large, often sparse, image. This implies that a pixel-wise classification model is needed for it to classify irregular shapes into the above mentioned classes. In the literature, this type of model is generally called a Semantic Segmentation model (Shelhamer et al., 2014), and is often used for similar tasks that require the identification of irregular shapes in a given image, be it for autonomous driving (Siam et al., 2018), medical applications (Asgari Taghanaki et al., 2021), or land cover segmentation (Ulmas & Liiv, 2020). Indeed, this problem has been around for a while (Liu et al., 2019), and the literature has converged towards using Fully Convolutional Networks (FCNs) that encode the input image similarly to a normal Convolutional Neural Network (CNN) architecture such as VGG-Net (Simonyan & Zisserman, 2015). This type of model was introduced by (Shelhamer et al., 2014) and, instead of having a fully connected layer at the end of the convolutional architecture, the semantic segmentation FCN model replaces this layer with an interpolation layer that converts each fully connected layer to a convolution. This last layer is then up-sampled by a "backward convolution", which then performs a pixel-wise classification, and outputs a segmentation of the input image.

Consequently, the FCN architecture laid the grounds for the current type of architecture used most frequently in semantic segmentation. Initially proposed by (Noh et al., 2015), the current basic segmentation architecture consists of two main blocks: an encoder, or convolution network, and a decoder, or deconvolution network. The encoder is, as mentioned before, a regular CNN architecture that encodes the image into a dense feature map. This feature map is then passed through the decoder, which is considered symmetrical to the encoder, but instead of performing convolutions, deconvolutions are performed to translate the dense feature map into a dense pixel-wise class prediction map. Indeed, the deconvolution operation is considered a convolution that, instead of translating wider feature maps into smaller ones, produces wider feature maps from smaller ones.

3.1.2. Baseline Architecture

From the basic architecture mentioned above, multiple variations are born in accordance to the need at hand. In our case, the architecture needed to solve the task of agriculture image segmentation needs to capture both large contexts and small details. For this, when the Agricultrue-vision dataset was initially introduced by (Chiu et al., 2020b), the DeepLabv3 and Deeplabv3+ architectures (Chen et al., 2017; 2018b) were the first architectures to be tested on the dataset. They achieved a test performance of 32.18% and 42.42% mIoU respectively. We will thus define the DeepLabv3+ as our *Baseline Architecture*.

The Baseline Architecture is based around the atrous convolution (Chen et al., 2018a), which instead of multiplying the feature map by a regular convolutional kernel, uses an atrous convolutional kernel that explictly adjusts the kernel's field of view on the feature map. A visualization provided by (Chen et al., 2018b) is shown in Figure 4. Therefore, they define their encoder as a ResNet (He et al., 2016) and freeze out the last feature map, where they apply Atrous Spatial Pyramid Pooling (ASPP), which convoludes this last feature map with atrous convolutional kernels of different fields of view, and then cocatenates the resultant feature maps into a single one that is fed to the decoder. The decoder concatenates the feature map outputted by the encoder with a low-level feature map from the initial layers of the ResNet in the decoder, and this is then up-sampled to output a prediction. The full model is illustrated in Figure 5.

Moreover, even though the DeepLabv3+ architecture was implemented quite recently on the Agriculture-vision dataset, better models have been encouraged by the 1st Agriculture-Vision Challenge (Chiu et al., 2020a) held between January and April 2020, where the best performing model achieve a mIoU of 63.9%. However, on this study we are going to only focus on the DeepLabv3+ architecture [2],

[2] We adapted the Pytorch implementation from https://github.com/VainF/DeepLabV3Plus-Pytorch to our particular task.

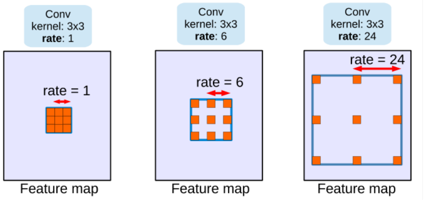

Figure 4. Atrous convolution proposed by (Chen et al., 2018a) showing different rates or kernel fields of view. The field of view varies directly with the rate, a parameter that can be adjusted to take into account more or less global features from the current feature map.

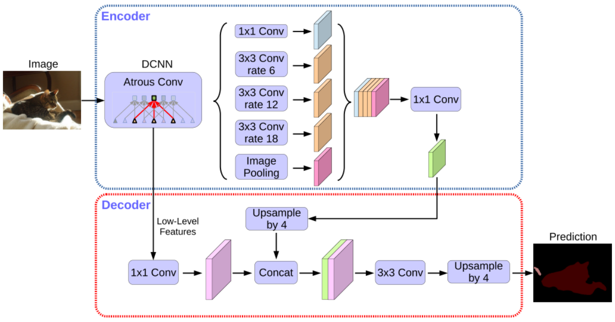

Figure 5. DeepLabv3+ model proposed by (Chen et al., 2017). In the encoder we can see the DeepLabv3 architecture (Chen et al., 2018b), and the decoder upsamples the feature map outputted by the encoder and concatenates it with a lower-level feature map from the encoder. This concatenated feature map is then upsampled to output the pixel-wise prediction.

since it has state-of-the-art performance in other semantic segmentation benchmarks, like Pascal VOC 2012 (Everingham et al., 2015) and Cityscapes (Cordts et al., 2016), and even in aerial segmentation tasks like Skyscapes (Azimi et al., 2019). Therefore, we propose to investigate the hypothesis that this architecture enhanced with some mechanisms to address the class imbalance and sparsity problems can achieve a strong performance on the Agriculture-Vision dataset.

3.2. Loss Functions

In order to determine the most appropriate loss function for this task, we consider different loss functions. As a baseline, we use the Cross-Entropy Loss which is commonly used in non-binary classification tasks as ours. Furthermore, we experiment with losses which have been designed to address class imbalance, such as the Focal Loss and Adaptive Class Weighting Loss and the Lovász-Softmax Loss, which is specifically crafted for semantic segmentation tasks.

The Focal Loss (Lin et al., 2017) corresponds to a dynamically scaled Cross-Entropy Loss in which the scaling factor decreases to zero as the model becomes more confident when predicting the correct class. This helps dealing with class imbalance as it allows the model to focus on hard examples and pay less attention to easy examples during training.

The Adaptive Class Weighting (ACW) Loss also addresses the class imbalance problem and was proposed by (Liu et al., 2020) in the 2020's Agriculture Vision competition. Rather than pre-computing fixed weights over the whole dataset, the authors proposed an adaptive class weighting method based on iterative batch-wise class rectification. To build their proposed loss function, this method is combined with a positive and negative class balanced function, rather than the cross-entropy function, and two regularisation terms.

The Lovász-Softmax Loss (Berman et al., 2018) is a loss function designed for multi-class semantic segmentation. It incorporates the softmax operation in the Lovász extension, which is a means that allows us to optimise the mIoU directly in neural networks.

3.3. Data Augmentation

The farmland patterns we aim to identify in our segmentation task are rotation-invariant, this is, if the image as a whole is rotated, the patterns in it are preserved. Therefore, to induce this rotation-invariance in our models, on every instance we augment our data with rotations. Following previous work with state-of-the-art results in this dataset (Liu et al., 2020) we first consider random 90°and 180°rotations in addition to vertical and horizontal flipping (which also preserve the patterns) as our baseline augmentations.

However, as an approach to solve the class imbalance problem mentioned in Section 2, we also consider additional augmentations on the two least represented classes ("Double Plant" and "Planter Skip"). The purpose of this is to provide our model with enhanced information of this classes, so that it can learn to recognise them better. For this reason, we experiment with a broader range of augmentations than the flipping and rotation applied as a baseline. Specifically, we consider three extra augmentation schemes:

1. **Aug-1**. In this first scheme, we add simultaneous translation, scaling, and random rotation of up to 45°.
2. **Aug-2**. Here, we adopt a more aggressive approach, and in addition to the baseline rotations and flipping and the transformations from Aug-1, we also perform random-size cropping, shift the RGB channels, and shuffle the channels.
3. **Aug-3**. In this scheme, in addition to the baseline flipping and rotations, we also perform random-size cropping.

Whether each augmentation is applied to a given input is determined at random with probabilities given in Appendix A.

3.4. Oversampling

As a related strategy to address the imbalance issue, we also perform oversampling of the less-represented classes during training. The rationale behind this is that during a given amount of iterations, the network can receive more frequently batches with inputs from the imbalanced classes. To achieve this, instead of sampling sequential batches of inputs, we sample them from an arbitrary distribution on the training set, where we artificially inflate the sampling probability of an image from a less represented class to be inversely proportional to its presence in the training set.

More specifically, we heuristically design weights for the images in the training set and then normalise those to obtain the probability of a given image being sampled for a batch. As mentioned in Section 2, the imbalance problem is pervasive at the pixel level, therefore we follow a pixel-wise approach to design the sample weight of each image. We assign an (unnormalised) weight to each class, c, denoted by q_c. Then, the sample weight, w_i of a given example i with a label mask, $\mathbf{y}_i$ is given by:

$$w_i = \sum_{c \in C} \sum_{n_c^i \in c} q_c * n_c^i$$

where n_c^i denotes the number of pixels from class c in the mask $\mathbf{y}_i$

Then, the probability of that particular example being sampled for a given mini-batch is the normalised weight for that example. We consider three different weighting schemes for the classes in which we assign higher weights to less represented classes, so that images with annotations from that classes can be sampled more frequently.

1. **Ovs-1**. In this scheme, we set $w_c = \frac{1}{6 s_c}$, where s_c is the percentage of images in the training set containing containing an annotation of class c.

2. **Ovs-2**. For this scheme, the weight from each class is given by: $w_c = \frac{1}{6 spx_c}$, where spx_c is the percentage of pixels (excluding background) annotated as class c.

3. **Ovs-3**. In this scheme, each class weight is given by $w_c = \dfrac{n_c}{\max_{c'} n_{c'}}$, where n_c is the total number of pixels annotated as class c. Hence, the most common class has a weight of 1, and the other weights are the ratio of the each class pixel frequency to that class.

4. Experiments

In this section we explore multiple possible solutions to the class-imbalance problem present in the Agriculture-Vision dataset with the goal of finding the model that best addresses the class imbalance problem present in the Double Plant and Planter Skip classes. For this, we start by exploring multiple output stride (os) values for the ASPP module in the DeepLabv3+ architecture on the full set, from which we choose the best os. With this os, we then downsample the images from a resolution of 512 × 512 to a resolution of 256 × 256 to speed up training. With this downsampled set, we explore the aforementioned Loss functions, Data Augmentations and Oversampling solutions. From here on, the mIoU metric is going to determine the best model, since it represents its overall performance. However, our qualitative analysis is going to focus on the IoU obtained for the two most imbalanced classes, and we will seek to maximise their IoU value.

4.1. Baselines

Initially, we compare the performance of the baseline architecture on the Agriculture-Vision dataset using different output strides. As mentioned in Section 2, we use mIoU as the evaluation metric, as well as the Intersection-over-Union (IoU) for each class.

We consider the Baseline Architecture as the DeepLabv3+ model with cross-entropy loss function and we subsequently train and validate it on three different output strides (8, 16 and 32). We trained the three models for 8100 iterations, amounting to 5 complete passes through the full dataset with learning rate of 0.01 and batch size of 8, using Stochastic Gradient Descent[3]. The results are presented in Table 1.

From these results we can see that the output stride of 32 gives a mIoU of 45.15%. Furthermore, we can see that the os 32 model outperforms the other output strides on 4 out of 7 of the classes, including the imbalanced and sparse classes, where the performance difference is significant, specially for the Planter Skip class. Since the output stride defines the rate of compression of the input feature map versus the output feature map, a bigger compression rate will mean that sparser annotations will be closer together, and therefore easier to classify. This is reflected in the obtained results, where an output stride of 32 produced better results in the sparser classes, specially in the Planter skip class. Therefore, from now on, we use an output stride of 32 and a down-sampled dataset with a resolution of 256 × 256 for the rest of the experiments.

4.2. Loss functions

For this part of the study we focus on experimenting with the multiple loss functions introduced in Section 4.2. We train 4 different models with the 4 different loss functions, Cross-entropy Loss, Adaptive Class Weighting Loss, Focal Loss, and Lovász-softmax Loss. From this study we seek to find the loss function that outputs the highest mIoU, while also outputting a significant improvement in the IoU for the Double Plant and Planter Skip classes. The results from these experiments can be found on Table 2.

From these results we can see that Adaptive Class Weighting outperforms the rest of the losses by 3 percentage points, approximately. This is thanks to a big improvement in the IoU of multiple classes, specially of the three most imbalanced classes: Double Plant, Planter Skip and Waterway. The more imbalanced the class, the more significant the improvement is, relative to the other loss functions. This is because the ACW loss dynamically updates the scaling weights of the loss function based on the pixel frequency at each training step and focuses on positive and negative

[3] The same hyperparameters will be kept for the rest of the experiments.

		Class						
Out. Stride	mIoU(%)	Background	Cloud shadow	Double plant	Planter skip	Standing water	Waterway	Weed cluster
(os=8)	44.99	**78.91**	45.21	23.61	0.19	57.53	63.05	46.45
(os=16)	44.72	78.75	43.70	21.45	0.34	**57.77**	**64.26**	46.73
(os=32)	**45.15**	78.00	**45.29**	**24.40**	**1.99**	57.46	61.69	**47.20**

Table 1. Baseline Architectures' validation mIoU performance on the full Agriculture-Vision dataset for different-sized output strides (os), which is the ratio at which the last feature map in the ResNet will be compressed compared to the input image. The smaller the os is, the denser the last feature map is. The mIoU for each class is also included, we can see that under-represented classes such as Planter Skip, Double Plant, and Waterway perform poorly.

samples, so instead of only computing the error for a single target label (the positive sample) out of the 7 classes, it computes the error taking into account all classes in the target label (the negative samples). This differs from the Cross-Entropy Loss, since this type of loss only focuses on the positive samples, and from the Focal Loss, as this function assigns weights based on the class' performance and also focuses only on positive samples. Something worth mentioning is that the Cross Entropy Loss, which is the one used for the baseline, still outperforms the Lovasz and Focal Loss, which were implemented with the intention of improving performance. However, in specially the Lovasz Loss, the model experimented a big drop in performance in the imbalanced class specially. This can be because, even though the Lovasz Loss function was chosen to help optimise the mIoU metric in our multi-class semantic segmentation task, it overlooks the class-imbalance and sparsity problems.

Furthermore, we plot the evolution of the IoU for the Double Plant and Planter Skip classes in Figure 6. From this plot we can visualise a significant performance boost given by the ACW loss.

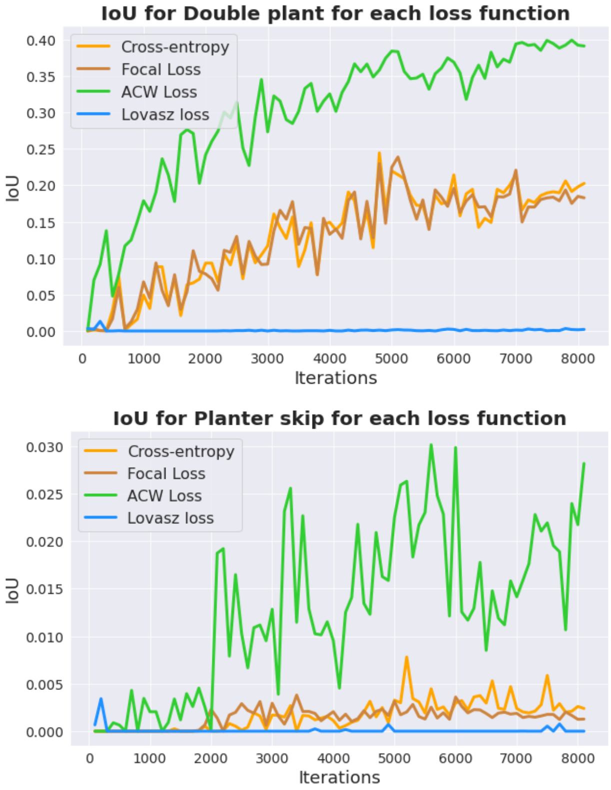

Figure 6. Intersection Over Union results for the two most imbalanced classes during training for 8100 iterations for the DeepLabv3+ model with os of 32 and different loss functions.

4.3. Data Augmentation

In this section we inspect how the different extra augmentation schemes from Section 3.3 complement the losses evaluated in the previous section to further enhance the performance on the imbalanced classes. We train the model with all the different combinations of loss functions and weight schemes, including the baseline augmentations. The mIoU results from each one of these can be found in Table 3.

From these results we can see that even though the ACW model still outperforms all the other ones, the data augmentations are actually detrimental to the performance. For all loss functions, and all augmentations, we find a decrease in performance with respect to the one with no augmentations. This decrease is significantly more severe with Aug-2, where we perform the most aggressive set of data augmentations. It seems that the performance decrease as we increase the level of augmentation is due to an underfitting of the model even in the classes with additional augmentation, as can be seen in Figure 7. The transformations to the data are actually making predictions more difficult thus hindering performance. This could potentially be fixed by training the model for a longer time, so it learns to make better predictions on the perturbed, added data.

4.4. Oversampling

For our last set of experiments, we implement the oversampling introduced in Section 4.4, where we try 3 different types of oversampling on the Cross Entropy Loss model (our Baseline Model with os 32), to finetune the weights from the oversampling. We choose the best performing oversampled model and combine this one with the ACW Loss. The results can be found in Table 4.

From these results we can see that the model with the best mIoU among the oversampling schemes is Ovs-3. This model however does not outperform the Cross Entropy Baseline Model, but it does outperform it in the imbalanced classes Double Plant and Planter Skip, as we can see in Figure 8. We also note that Ovs-2 outperforms the rest of the oversampling schemes and the baseline on these two classes. However, since the model that outputs the best mIoU from the oversampling schemes is model Ovs-3, we choose this one to train it with ACW loss. Indeed, this

		Class						
Loss Fn.	mIoU(%)	Background	Cloud shadow	Double plant	Planter skip	Standing water	Waterway	Weed cluster
Cross Entr.	45.45	79.31	50.89	19.79	0.26	**59.45**	61.81	46.65
Focal	45.33	**79.63**	**51.58**	18.48	0.12	58.65	61.70	47.15
ACW	**48.72**	78.57	46.87	**39.12**	**2.81**	58.94	**65.89**	**48.86**
Lovasz	35.33	78.02	44.12	0.13	0.00	45.73	44.89	34.40

Table 2. Results from experimenting with multiple loss functions to address the class imbalance problem present in the dataset. Adaptive Class Weighting outperforms the rest of the losses by 3 percentage points, offering significant improvements specially in the imbalanced classes.

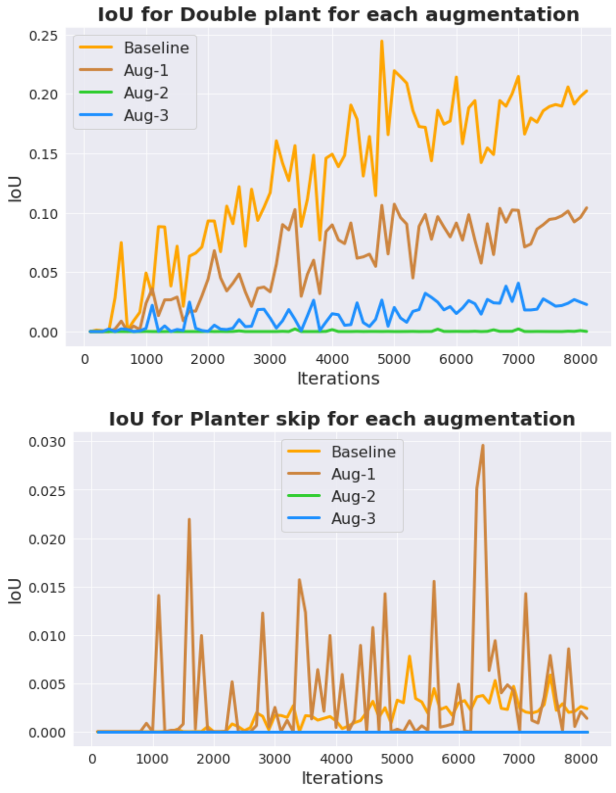

Figure 7. Intersection Over Union results for the two most imbalanced classes during training for 8100 iterations for the DeepLabv3+ model with os of 32 and the Cross-Entropy Loss function with different augmentation schemes.

Loss Fn.	No Aug	Aug-1	Aug-2	Aug-3
Cross Entr.	45.45	43.98	42.50	42.68
Focal	45.33	43.24	37.86	42.15
ACW	**48.72**	**46.21**	**43.34**	**44.64**
Lovasz	35.33	35.16	42.50	36.09

Table 3. Augmentations results (mIoU (%)) for the 3 augmentations introduced in Section 4.3, and no augmentation at all, for the 4 different losses evaluated in the previous section. We can see that the ACW loss still outperforms the rest of the models on all augmentations.

model outperforms all other models from this experiment including the Baseline. It also significantly improves the performance of imbalanced class Double Plant, but the weights actually compromise performance from the Background, Cloud Shadow, Planter Skip, and Standing Water classes. However, this oversampling scheme still resulted in a worse mIoU than when no oversampling was used.

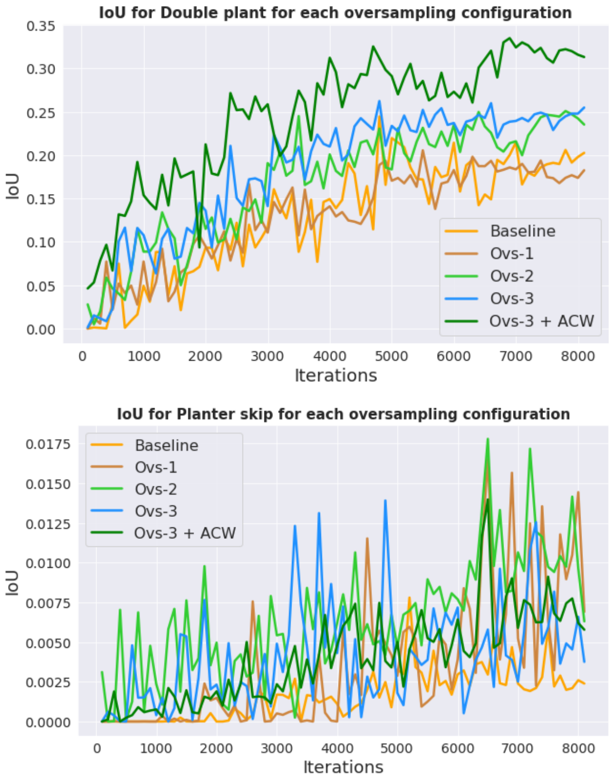

Figure 8. Intersection Over Union results for the two most imbalanced classes during training for 8100 iterations for the DeepLabv3+ model with os of 32 and the Cross-Entropy Loss function with different oversampling weighting schemes.

The oversampling work was expected to improve the performance in the more sparse classes (Double Plant and Planter Skip), especially Ovs-2 where the class weights were based on their pixel frequency. However, this was at the cost of degraded performance on most-represented classes at the pixel level (Cloud Shadow and Background) which led to an overall worse mIoU. It is important to note that for all the experiment we trained for the same number of iterations as in the baseline; hence, oversampling a given class necessarily must be done at the expense of undersampling some other. Therefore, we would expect that training for a longer number of iterations would improve the performance on the undersampled classes and with it the overall mIoU.

		Class						
Oversampling	mIoU(%)	Background	Cloud shadow	Double plant	Planter skip	Standing water	Waterway	Weed cluster
Baseline (No Ovs)	45.45	**79.31**	**50.89**	19.79	0.26	59.45	61.81	46.65
Ovs-1	44.32	75.39	48.76	20.55	0.13	60.40	57.84	47.19
Ovs-2	44.79	76.00	41.91	24.71	**1.41**	**60.76**	61.86	46.87
Ovs-3	44.97	78.33	46.29	24.67	0.51	57.95	60.98	46.06
Ovs-3 + ACW	**45.67**	75.59	42.79	**31.31**	0.58	58.78	**62.39**	**48.27**

Table 4. Oversampling results for the 3 oversampling weights introduced in Section 4.4 and implemented on the Cross Entropy Loss Baseline Model. From these results we determine that the best oversampling weights are from Oversampling 3, and thus we implement ACW Loss with these weights.

		Class						
Model	mIoU(%)	Background	Cloud shadow	Double plant	Planter skip	Standing water	Waterway	Weed cluster
Validation (Baseline)	44.99	78.91	45.21	23.61	0.19	57.53	63.05	46.45
Validation (Best)	49.42	78.78	50.00	43.16	5.81	54.32	65.04	48.84
Test (Best)	46.30	77.61	37.17	28.54	22.94	57.79	55.24	46.78

Table 5. Validation and test results of our Final Model compared to the Baseline Model. Our model improves the performance of the Baseline Model by addressing the class-imbalance problem specially on the Double Plant and Planter Skip classes.

4.5. Best model

Finally, from all the sets of experiments we find that the best model was the ACW Loss model without any augmentations or oversampling. Indeed, if we look at Table 2, we find that this model has an mIoU of 48.72% outperforming all models from all experiments. The only two models that come close in terms of performance are the model with Augmentation 1 and ACW Loss from Section 4.3, with an mIoU of 46.21%, and the Oversampling 3 + ACW Loss model from Section 4.4, with an mIoU of 45.67%. And these latter models seem to only degrade the performance of the ACW Loss model without any augmentations or oversampling.

We thus decide to train the ACW Loss model, which we will call the Final Model from now on. The Final Model was trained on the original-sized images (so 512 × 512) for 8100 iterations keeping the rest of the hyperparameters the same. We then use the trained model and test it on the test images, whose output was then submitted to the CodaLab server for the Agriculture-Vision challenge.

We thus obtain the results found in Table 5. From these results we can see that we are able to outperform the Baseline implementation, with an increase in validation mIoU of approximately 5 percentage points. Additionally, if we compare the Validation results of the Final Model with its test results, we can see a big boost in the IoU of the most imbalalanced classes. These test results are also compared to last year's leaderboard on Appendix B, where we rank 25 out of 71 teams (The table shows the top 40 teams).

5. Conclusions

In this study we focus on improving the performance of the baseline DeepLabv3+ model on the Agriculture-Vision semantic segmentation task. We do this by specifically addressing the class imbalance problem present on the training set using separate methods that have been found to aid models perform better on imbalanced datasets. The methods we focus on are (1) using multiple loss functions, (2) performing data augmentation, and (3) performing oversampling on the data.

We train multiple models that implement these methods to inspect their impact on the performance of the Baseline model. After experimenting on all of these models, we found that adding Adaptive Class Weighting Loss to the Baseline output the best performance and, contrary to what we expected, none of the data augmentation or oversampling schemes gave a significant performance improvement. Our final model achieves an mIoU of 46.40% on the test set, and would achieve a 25 out 71 ranking in last year's Agriculture-Vision competition.

In future work, we could focus on addressing the sparsity problem present in the data. This could be done by experimenting with more sophisticated architectures that can represent global features while being able to capture more detailed information. Alternatively, we could incorporate some domain knowledge about agriculture imaging to the task, similar to what Sheng et al. (2020) did with various vegetation indices.

Another approach worthy of attempting could be to use adaptive oversampling weights or more principled approaches to perform augmentation and oversampling on imbalanced datasets. For instance, one direction could be to adapt for image segmentation the SMOTE technique (Chawla et al., 2002) which is commonly used on imbalanced classification tasks.

Finally, another way of addressing the class-imbalance problem can be by using Incremental Transfer Learning (Abdou et al., 2018), which consists in training our model in stages, beginning with the least represented class, and progressively adding the next-least-represented class after each iteration, until all classes are represented (Sander, 2020).

References

Abdou, Mohammed, Elkhateeb, Mahmoud, Sobh, Ibrahim, and El-sallab, Ahmad. Weighted self-incremental transfer learning for 3d-semantic segmentation. 2018.

Asgari Taghanaki, Saeid, Abhishek, Kumar, Cohen, Joseph Paul, Cohen-Adad, Julien, and Hamarneh, Ghassan. Deep semantic segmentation of natural and medical images: a review. Technical Report 1, 2021.

Azimi, Seyed Majid, Henry, Corentin, Sommer, Lars, Schumann, Arne, and Vig, Eleonora. Skyscapes fine-grained semantic understanding of aerial scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7393–7403, 2019.

Berman, Maxim, Triki, Amal Rannen, and Blaschko, Matthew B. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4413–4421, 2018.

Chawla, Nitesh V, Bowyer, Kevin W, Hall, Lawrence O, and Kegelmeyer, W Philip. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Chen, Liang Chieh, Papandreou, George, Schroff, Florian, and Adam, Hartwig. Rethinking atrous convolution for semantic image segmentation. *arXiv*, 2017. ISSN 23318422.

Chen, Liang Chieh, Papandreou, George, Kokkinos, Iasonas, Murphy, Kevin, and Yuille, Alan L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 4 2018a. ISSN 01628828. doi: 10.1109/TPAMI.2017.2699184. URL http://liangchiehchen.com/projects/.

Chen, Liang-Chieh, Zhu, Yukun, Papandreou, George, Schroff, Florian, and Adam, Hartwig. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018b.

Chiu, Mang Tik, Xu, Xingqian, Wang, Kai, Hobbs, Jennifer, Hovakimyan, Naira, Huang, Thomas S., Shi, Honghui, Wei, Yunchao, Huang, Zilong, Schwing, Alexander, Brunner, Robert, Dozier, Ivan, Dozier, Wyatt, Ghandilyan, Karen, Wilson, David, Park, Hyunseong, Kim, Junhee, Kim, Sungho, Liu, Qinghui, Kampffmeyer, Michael C., Jenssen, Robert, Salberg, Arnt B., Barbosa, Alexandre, Trevisan, Rodrigo, Zhao, Bingchen, Yu, Shaozuo, Yang, Siwei, Wang, Yin, Sheng, Hao, Chen, Xiao, Su, Jingyi, Rajagopal, Ram, Ng, Andrew, Huynh, Van Thong, Kim, Soo Hyung, Na, In Seop, Baid, Ujjwal, Innani, Shubham, Dutande, Prasad, Baheti, Bhakti, Talbar, Sanjay, and Tang, Jianyu. The 1st agriculture-vision challenge: Methods and results. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020-June:212–218, 2020a. ISSN 21607516. doi: 10.1109/CVPRW50498.2020.00032.

Chiu, Mang Tik, Xu, Xingqian, Wei, Yunchao, Huang, Zilong, Schwing, Alexander, Brunner, Robert, Khachatrian, Hrant, Karapetyan, Hovnatan, Dozier, Ivan, Rose, Greg, Wilson, David, Tudor, Adrian, Hovakimyan, Naira, Huang, Thomas S., and Shi, Honghui. Agriculture-Vision: A Large Aerial Image Database for Agricultural Pattern Analysis. *arXiv*, pp. 2828–2838, 2020b. ISSN 23318422.

Cordts, Marius, Omran, Mohamed, Ramos, Sebastian, Rehfeld, Timo, Enzweiler, Markus, Benenson, Rodrigo, Franke, Uwe, Roth, Stefan, and Schiele, Bernt. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. pp. 248–255. Institute of Electrical and Electronics Engineers (IEEE), 3 2010. doi: 10.1109/cvpr.2009.5206848.

Everingham, Mark, Eslami, SM Ali, Van Gool, Luc, Williams, Christopher KI, Winn, John, and Zisserman, Andrew. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, pp. 770–778. IEEE Computer Society, 12 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.90. URL http://image-net.org/challenges/LSVRC/2015/.

Kamilaris, Andreas and Prenafeta-Boldú, Francesc X. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.

Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross, He, Kaiming, and Dollár, Piotr. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Liu, Qinghui, Kampffmeyer, Michael, Jenssen, Robert, and Salberg, Arnt Borre. Multi-view self-constructing graph convolutional networks with adaptive class weighting loss for semantic segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020-June(0314):199–205, 2020. ISSN 21607516. doi: 10.1109/CVPRW50498.2020.00030.

Liu, Xiaolong, Deng, Zhidong, and Yang, Yuhan. Recent progress in semantic image segmentation. *Artificial Intelligence Review*, 52(2):1089–1106, 2019. ISSN

15737462. doi: 10.1007/s10462-018-9641-3. URL https://doi.org/10.1007/s10462-018-9641-3.

Noh, Hyeonwoo, Hong, Seunghoon, and Han, Bohyung. Learning deconvolution network for semantic segmentation. Technical report, 2015.

Sander, Ryan. Sparse data fusion and class imbalance correction techniques for efficient multi-class point cloud semantic segmentation, 02 2020.

Shelhamer, Evan, Long, Jonathan, and Darrell, Trevor. Fully Convolutional Networks for Semantic Segmentation. Technical report, 2014.

Sheng, Hao, Chen, Xiao, Su, Jingyi, Rajagopal, Ram, and Ng, Andrew. Effective data fusion with generalized vegetation index: Evidence from land cover segmentation in agriculture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 60–61, 2020.

Siam, Mennatullah, Elkerdawy, Sara, Jagersand, Martin, and Yogamani, Senthil. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. Technical report, 2018. URL https://www.researchgate.net/publication/318336948.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 9 2015. URL http://www.robots.ox.ac.uk/.

Ulmas, Priit and Liiv, Innar. Segmentation of satellite imagery using U-Net models for land cover classification. *arXiv*, 3 2020. ISSN 23318422. URL http://arxiv.org/abs/2003.02899.

A. Augmentations

Transformation	Baseline	Aug-1	Aug-2	Aug-3
Vertical Flip	0.5	0.5	0.5	0.5
Horizontal Flip	0.5	0.5	0.5	0.5
Random 90 Rotation	0.5	0.5	0.5	0.5
Translation+Scale+Rotation	-	1.0	1.0	-
Random Size Crop	-	-	1.0	1.0
RGB Shift	-	-	1.0	-
Channel Shuffle	-	-	0.25	-

Table 6. Probabilities of each transformation for the different augmentation schemes.

B. Challenge Leaderboard

Team	mIoU (%)	Background	Cloud shadow	Double plant	Planter skip	Standing water	Waterway	Weed cluster
Hyunseong	63.9	80.6	56.0	57.9	57.5	75.0	63.7	56.9
seungjae	62.2	79.3	44.4	60.4	65.9	76.9	55.4	53.2
yjl912.2	61.5	80.1	53.7	46.1	48.6	76.8	71.5	53.6
ddcm	60.8	80.5	51.0	58.6	49.8	72.0	59.8	53.8
RodrigoTrevisan	60.5	80.2	43.8	57.5	51.6	75.3	66.2	49.2
SYDu	59.5	81.3	41.6	50.3	43.4	73.2	71.7	55.2
agri	59.2	78.2	55.8	42.9	42.0	77.5	64.7	53.2
Tennant	57.4	79.9	36.6	54.8	41.4	69.8	66.9	52.0
celery03.0	55.4	79.1	38.9	43.3	41.2	73.0	61.5	50.5
stevenwudi	55.0	77.4	42.0	54.4	20.1	69.5	67.7	53.8
PAII	55.0	79.9	38.6	47.6	26.2	74.6	62.1	55.7
agrichallenge1.2	54.6	80.9	50.9	39.3	29.2	73.4	57.8	50.5
hui	54.0	80.2	41.6	46.4	20.8	72.8	64.8	51.4
shenchen61.6	53.7	79.4	36.7	56.3	21.6	67.0	61.8	52.8
NTU	53.6	79.8	41.4	49.4	13.5	73.3	61.8	56.0
tpys	53.0	81.1	50.5	37.1	25.9	67.4	58.7	50.1
Simple	52.7	80.2	40.0	45.2	24.6	70.9	57.6	50.4
Ursus	52.3	78.9	36.3	37.8	34.4	69.3	57.1	52.3
liepieshov	52.1	77.2	40.2	46.0	16.0	71.3	62.9	51.1
Lunhao	49.4	79.5	40.4	38.8	10.5	69.4	58.3	49.1
tetelias-mipt	49.2	80.4	37.8	34.8	04.6	70.6	62.5	53.8
Dataloader	48.9	79.1	42.0	35.8	09.1	68.7	56.7	51.3
Hakjin	46.4	78.6	32.0	38.3	01.8	66.2	58.0	49.9
G012 (ours)	**46.3**	**77.6**	**37.2**	**28.6**	**22.9**	**57.8**	**55.2**	**46.8**
JianyuTANG	44.6	78.1	37.9	31.8	15.4	47.3	54.8	46.9
haossr	43.9	79.2	21.4	28.1	02.7	67.5	56.4	52.3
rpartsey	41.5	72.5	21.6	36.2	09.1	59.7	40.7	50.6
baidujjwal	40.8	75.2	26.1	40.1	09.9	48.0	37.1	49.5
Chaturlal	40.7	77.7	23.0	20.4	05.0	55.0	51.0	52.9
Sciforce	40.2	80.5	29.6	24.4	0.0	41.2	55.9	50.0
MustafaA	40.1	76.5	34.4	25.6	11.1	46.0	36.5	50.3
HaotianYan	36.8	77.1	21.9	25.1	13.7	57.5	24.3	37.9
gro	36.3	76.4	37.5	08.4	0.0	60.3	29.7	41.8
oscmansan	35.5	71.6	29.6	03.0	0.0	52.4	46.2	45.9
ThorstenC	33.6	72.3	22.3	10.0	02.0	40.8	40.1	47.8
ZHwang	33.5	76.5	32.4	12.9	0.0	57.2	15.9	39.9
fayzur2.0	22.1	65.4	21.8	02.2	00.2	23.3	13.4	28.7
gaslen.2	21.5	71.0	03.3	17.9	00.8	10.2	06.9	40.1
dvkhandelwal	16.3	71.5	0.0	0.0	0.0	42.6	0.0	0.0
ajeetsinghiitd	10.3	56.9	00.2	00.4	0.0	0.0	00.1	14.5

Table 7. 1st Agriculture-Vision Prize Challenge leaderboard with our results.

C. Comparison of prediction results of the Baseline versus the Final model

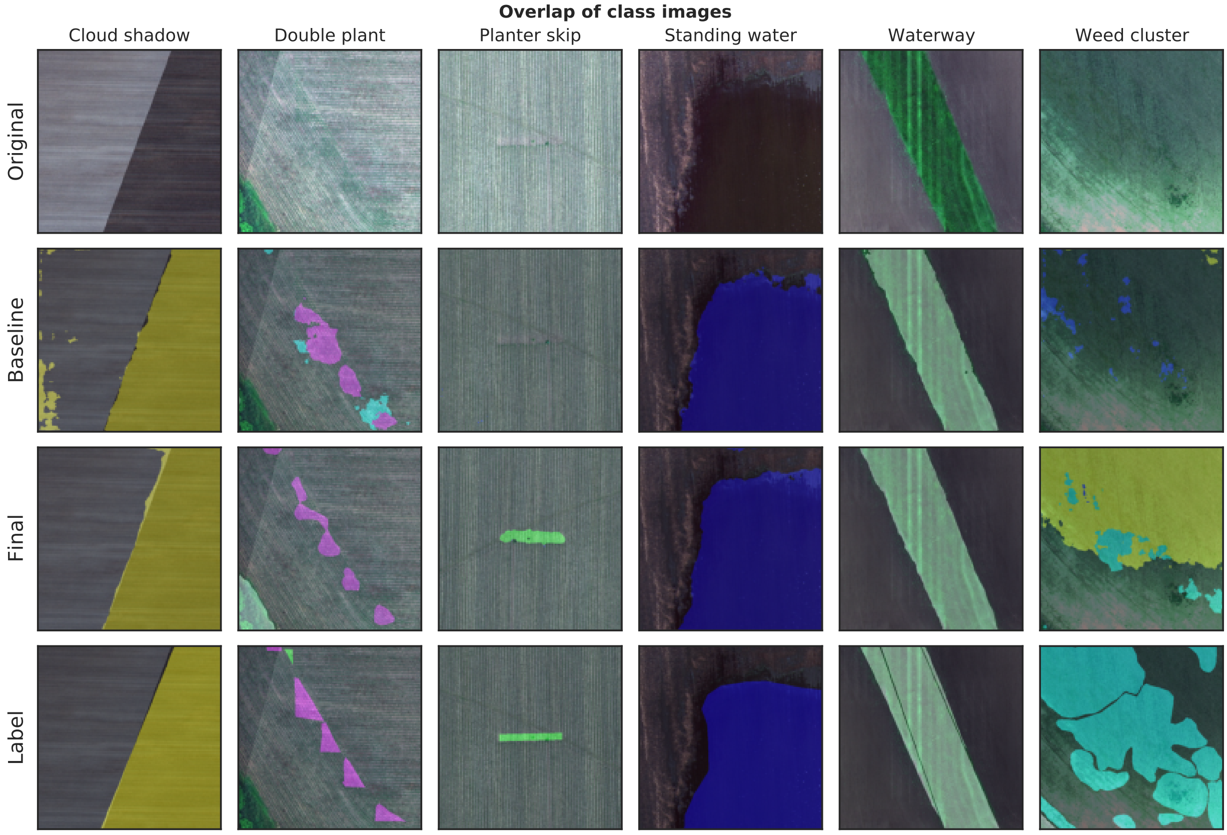

Table 8. Matrix of image predictions for each class in the dataset comparing the original image to the Baseline prediction, the Final Model prediction and the ground truth (Label). From this figure we can see that the Final model performs better at making a prediction for Double plant and Planter skip.