

# Improving Data Efficiency for Plant Cover Prediction with Label Interpolation and Monte-Carlo Cropping

Matthias Körschens<sup>1,2</sup>[0000–0002–0755–2006], Solveig Franziska Bucher<sup>1,2,3</sup>[0000–0002–2303–4583], Christine Römermann<sup>1,2,3</sup>[0000–0003–3471–0951], and Joachim Denzler<sup>1,2,3</sup>[0000–0002–3193–3300]

<sup>1</sup> Friedrich Schiller University, D-07743 Jena, Germany

<sup>2</sup> German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, D-04103 Leipzig, Germany

<sup>3</sup> Michael Stifel Center Jena, D-07743 Jena, Germany  
`{matthias.koerschens,solveig.franziska.bucher,  
christine.roemermann,joachim.denzler}@uni-jena.de`

**Abstract.** The plant community composition is an essential indicator of environmental changes and is, for this reason, usually analyzed in ecological field studies in terms of the so-called plant cover. The manual acquisition of this kind of data is time-consuming, laborious, and prone to human error. Automated camera systems can collect high-resolution images of the surveyed vegetation plots at a high frequency. In combination with subsequent algorithmic analysis, it is possible to objectively extract information on plant community composition quickly and with little human effort. An automated camera system can easily collect the large amounts of image data necessary to train a Deep Learning system for automatic analysis. However, due to the amount of work required to annotate vegetation images with plant cover data, only few labeled samples are available. As automated camera systems can collect many pictures without labels, we introduce an approach to interpolate the sparse labels in the collected vegetation plot time series down to the intermediate dense and unlabeled images to artificially increase our training dataset to seven times its original size. Moreover, we introduce a new method we call Monte-Carlo Cropping. This approach trains on a collection of cropped parts of the training images to deal with high-resolution images efficiently, implicitly augment the training images, and speed up training. We evaluate both approaches on a plant cover dataset containing images of herbaceous plant communities and find that our methods lead to improvements in the species, community, and segmentation metrics investigated.

**Keywords:** Convolutional Neural Networks · Plant Cover Prediction · Ecology · Biodiversity Monitoring · Small Data · Monte-Carlo · Time Series.

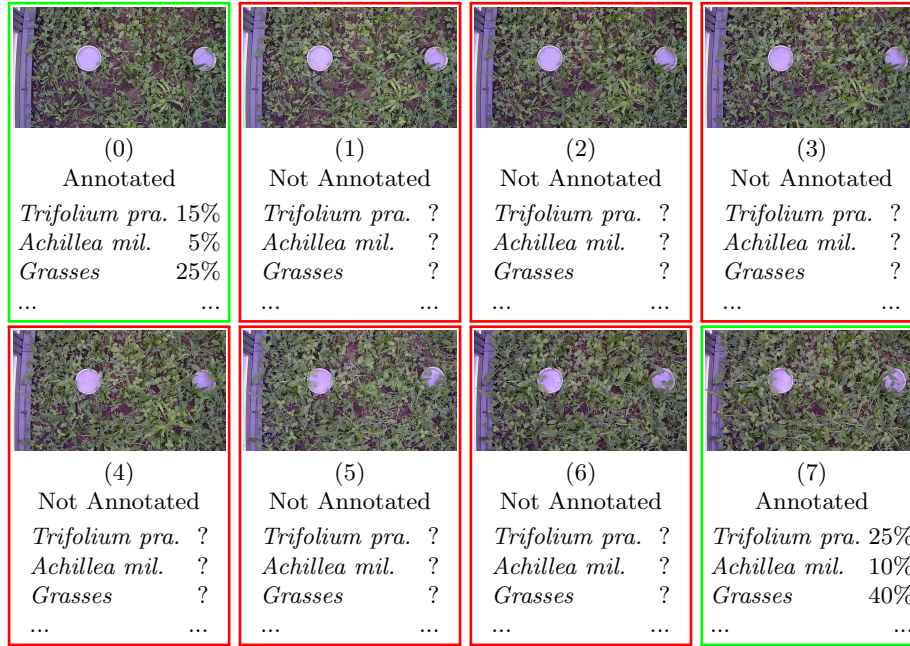


Fig. 1: From the entire dataset, since there are weekly annotations, but daily images, only about one in seven images is annotated. While the images do not significantly differ from one day to the next, the small differences between the images can still help the model learn intermediate growth stages of the plants.

## 1 Introduction

The plant community composition is an essential indicator for environmental changes such as changes in climate change [20,18,19], insect abundance [23,24], and land-use [7,9]. Hence, this kind of data is usually collected by plant ecologists [18,7,23,3], for example, in the form of measuring the plant cover, in regular, but rather long, time intervals. The plant cover is defined as the percentage of area of ground covered by each plant species disregarding any occlusion. Usually, many different plant species are contained in a single plot, which often overgrow and occlude each other, making the estimation of the plant cover a very complex task.

Automated plant cover prediction can be a vital asset to plant biodiversity researchers. Abundance values, like the plant cover, are traditionally collected manually by estimating them directly in the field on vegetation plots by visual inspection (see Figure 1). However, collecting data this way is laborious, prone to human error, and subjective. Therefore, automated systems performing such estimations offer a significant advantage to these traditional methods, as they can analyze a large number of images of such vegetation plots in a short amount of time and deliver valuable research data at a high temporal resolution. The

collected plant abundance data can then be used to determine the influence of environmental changes on the plant communities. The high temporal resolution of the automatically extracted data offers the potential for very fine-grained analyses, such that a shift in the community distribution can be investigated in intervals of days or even hours instead of only weeks [24], months [1] or years [9].

To establish an automated system to perform such an analysis of images, convolutional neural networks (CNNs) are a good choice, as they are powerful image processing models. However, they usually require large amounts of labeled training data to perform well. Körschens *et al.* [13] demonstrated a way to determine the plant cover by training on the so-called InsectArmageddon dataset [24,13], which we will also investigate in this work and which contains merely 682 labeled images, collected and annotated with plant cover estimates in weekly intervals. As this number of training images is relatively low, especially in conjunction with such a complex task, the quality of the results is very likely limited by the amount of available training data.

To solve the issue of little labeled data, we investigate using unlabeled intermediate images to increase the size of the training set. Plant cover estimates for vegetation plots are very laborious to create. However, additional unlabeled images are not. If an automated camera system exists to gather images for training or automatic analysis of plant cover, it can usually also collect a large number of additional unlabeled images at almost no cost. In the InsectArmageddon dataset, images are collected at a daily basis but only annotated at the weekly one. We leverage this experimental setup to automatically generate weak labels for the intermediate days between two days for which human annotations are given. The key idea is to handle uncertainty in the weak labels by weighting them according to their temporal distance to the next reference estimate. We will refer to this approach as label interpolation.

In addition to this, to enable the network to train on images at their full resolution, we propose a Monte-Carlo sampling approach for training the network, which we will refer to as Monte-Carlo Cropping (MCC). The original image is sampled in equally-sized patches, for each of which the target output is estimated individually. Afterwards, the network output for all patches sampled from a single image is averaged. This kind of sampling empirically seems to have a regularizing effect on the network training, leading to better results on high-resolution images and drastically reducing the training time of such images.

In the following, we will elaborate on related work to our approaches, followed by a detailed explanation of our methods, experimental results, and finally, a conclusion.

## 2 Related Work

*Label Interpolation.* In this work, we investigate the problem of utilizing unlabeled images for training in addition to a small number of images with labels. This problem is usually tackled by semi-supervised learning approaches, espe-

cially self-training [22,11]. In self-training methods, a model is trained with few annotated images and then used to label available but unannotated images to increase the size of the training set iteratively. In contrast to these approaches, label interpolation heavily utilizes the strong correlation of plant cover values in the time series to generate labels and does not rely on trained models at all. Similarly, the data augmentation method mixup [27] also interpolates labels to generate novel annotations. However, in contrast to our method, the authors do not apply the new labels to unlabeled images but fuse two existing images and their class labels.

*Monte-Carlo Cropping (MCC)*. Another problem we tackle in this work is the utilization of high-resolution images in CNN training. Cropping the original training images into much smaller images is a simple approach to this problem, and is usually applied in tasks like image segmentation and object detection, that also often deal with high-resolution images [25,26,4,16,15]. For these tasks, cropping is usually done a single time per epoch per image, and the ground-truth data is also adapted in the same way. For image segmentation, the ground-truth data are usually segmentation maps, which have the same dimensions as the original image, and can also be cropped in the same way. The ground-truth for object detection are usually bounding box coordinates in the original images, which can also be easily be adapted to the cropped input image by systematically modifying the coordinates. For plant cover estimation, however, the target data are merely numerical vectors representing the plant cover distribution in the image, and can therefore neither be cropped or simply adapted. To solve this problem, our Monte-Carlo Cropping introduces a stochastical component in order to be able to approximate the underlying plant cover distribution, which does not need to be done for image segmentation and object detection.

### 3 Methods

#### 3.1 Label Interpolation

The first method we introduce is label interpolation. As shown in Figure 1, from seven existing images in a single week, only a single one is labeled, leaving the other images unused. Moreover, we can see that the differences between the daily pictures are only minor compared to the weekly differences; however not insignificant. Images of this kind have two advantages. Firstly, the network can learn the growth process of plants in much more fine-grained steps, especially since this kind of data contains more and new information compared to simple augmented images. And secondly, since the differences between the pictures are relatively small, we can infer certain properties of the supposed labels for these images from their neighboring annotations. More formally, for our label interpolation method to work, we take advantage of the fact that plant cover estimates are continuous values. Moreover, we assume that the intermediate value theorem [2] holds for these estimates collected in a time series. That is, if a plant’s measured

cover value in a certain week was  $\text{cover}(t_0)$  and  $\text{cover}(t_1)$  in the following week,

$$\forall v \text{ with } \min(\text{cover}(t_0), \text{cover}(t_1)) < v < \max(\text{cover}(t_0), \text{cover}(t_1)), \quad (1)$$

$$\exists t \in (t_0, t_1) : \text{cover}(t) = v. \quad (2)$$

Under the assumption that plants grow in a continuous fashion without external interference, this theorem holds.

Here, we will utilize linear interpolation, specifically of the data of two subsequent weeks, which implicitly weights the values respective to their temporal distance to the next annotated data point:

$$\text{cover}(t) = \frac{\text{cover}(t_0)(t_1 - t) + \text{cover}(t_1)(t - t_0)}{t_1 - t_0}. \quad (3)$$

A linear interpolation might not precisely represent the growth process of the plants and discontinuities in the images (like occlusion). However, with such small time steps, the growth process of the plants can be assumed to be approximately linear between two weeks. We are aware of violations of our assumptions in practice. However, empirically such issues play only a minor role when it comes to the overall quality of our suggested approach.

### 3.2 Monte-Carlo Cropping

The second method we introduce here is Monte-Carlo Cropping (MCC). A significant problem in plant cover prediction is that the images provided by the camera systems usually have a relatively high resolution (e.g.,  $2688 \times 1520$  pixels for the InsectArmageddon dataset). However, networks are usually only applied on rather small, often downsampled images ( $224 \times 224$  for typical ImageNet [21] tasks, and  $448 \times 448$  or similar for fine-grained ones [5]).

Training on large images is computationally expensive, consumes large amounts of memory, and can take a long time. For the original image  $I \in \mathbb{R}^{H \times W \times 3}$  we sample patches  $P \in \mathbb{R}^{h \times w \times 3}$  with  $h \ll H$  and  $w \ll W$ . For each patch  $P$ , we let the network predict the plant cover separately and then average these values over the number of patches sampled from each image.

Since the patches are sampled from an image with the plant cover values  $\text{cover}_p$ , the expected value is equal to  $\text{cover}_p$  for a large number of patches sampled. Therefore, due to the law of large numbers [6], the following holds:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \text{cover}_{i,p} = \text{cover}_p, \quad (4)$$

with  $n$  being the number of patches sampled,  $i$  being the index of the randomly sampled patch, and  $p$  denoting plant species. I.e., while the plant cover values observed in the smaller patches do not necessarily reflect the values of the total images when selecting a sufficiently large number of patches, they do so on average.



Fig. 2: An example of a random patch selection with Monte-Carlo Cropping.

During training, we sample equally sized square patches from the original image, an example of which can be seen in Figure 2. It is visible that the number of pixels shown to the network is significantly reduced, depending on the size of the patches and number of patches sampled. Hence, computational complexity can also be reduced with MCC.

## 4 Experimental Results

### 4.1 Dataset

In our experiments, we utilize the InsectArmageddon dataset [24,14] from the eponymous iDiv project that took place in 2018 over multiple months. The images were collected in 24 so-called EcoUnits, which are boxes containing small enclosed ecosystems. Each of the EcoUnits was equipped with two cameras that collected daily pictures of these ecosystems. The dataset from the InsectArmageddon experiment comprises estimated plant cover data (“reference estimates”) for eight herbaceous plant species in 682 images collected weekly by a single ecologist. The image set with original and interpolated annotations contains about 4900 images, i.e., about seven times the number of images due to one originally labeled image per week, and six with interpolated labels. On this dataset, we perform 12-fold cross-validation by selecting the images of two EcoUnits for testing, and the ones of the remaining 22 EcoUnits for training. For more details on the InsectArmageddon image dataset, we would like to refer to [24] and [14].

### 4.2 Setup

We use the approach introduced in [13]. i.e., we utilize a ResNet50 [8], architecture with Feature Pyramid Network [17], the 3-phase pre-training pipeline based on freely available images from GBIF<sup>4</sup> to train a classification network (phase 1), which generates simple segmentations with class activation mapping (CAM) [28], on which we then pre-train a segmentation network (phase 2). The weights

<sup>4</sup> <http://gbif.org>

of this network are then used as initialization for our plant cover prediction network, which we train on the plant cover annotations (phase 3).

During the first phase, we utilize global log-sum-exp-pooling [12,13] with a learning rate of  $10^{-4}$  and a weight decay of  $10^{-4}$ . We use this pooling method, as in [12] it generated better segmentations when used in conjunction with CAM. Moreover, we use a categorical cross-entropy loss during optimization and train with early stopping and dynamic learning rate reduction. The learning rate is reduced by a factor of 10 when there is no improvement in the validation accuracy over four epochs, and the training is stopped, if there is no improvement over six epochs. In the second phase, we use a learning rate of  $10^{-5}$ , a weight decay of  $10^{-4}$ , and a combination of binary-cross-entropy and dice loss, which are summed up and weighted equally as loss. During this training, we also used a dynamic learning rate adaptation and early stopping; however, we monitored the mean Intersection over Union (mIoU) instead of the accuracy. In the third phase, we train with a batch size of 1 and a learning rate of  $10^{-5}$ , which is reduced by a factor of 10 after 50% and 75% of the total epochs, respectively. We are using the mean scaled absolute error ( $MSAE_{\sigma}$ ) as loss. This error is defined as

$$MSAE_{\sigma}(\mathbf{t}, \mathbf{p}) = \frac{1}{n} \sum_{i=1}^n \left| \frac{t_i}{\sigma_i} - \frac{p_i}{\sigma_i} \right|, \quad (5)$$

where  $\sigma$ , in our case, is the standard deviation of the species-wise plant cover values calculated over the training dataset. This loss aims to reweight the species to account for the substantial imbalance in the dataset.

In our experiments, we compare the image resolution used in previous works [13] ( $1536 \times 768$  pixels) and the full image resolution ( $2688 \times 1536$  pixels). We investigate several training durations and their effect on the training with weekly and interpolated daily images. For the daily labels we investigate fewer epochs, since the iteration count per epoch is much higher in comparison to the weekly label dataset.

For our experiments on MCC, we investigate different patch sizes as well as several patch counts. We chose the patch sizes of  $128^2$ ,  $256^2$ ,  $512^2$ , and  $1024^2$  pixels, and for each patch size, a respective sample count so that the number of pixels sampled is around 50% of the pixel count of the original image. For each sample count selected, we also investigate a pixel count of half or double the number of pixels sampled. For a patch of  $512^2$  pixels, we investigate sample counts of 8, 4, and 16; for a patch of  $256^2$  pixels, sample counts of 32, 16, and 64, etc. It should be noted that the cropped image patches are input into the network as-is without any additional resizing.

### 4.3 Metrics

We investigate three different metrics to analyze our results. The first metric is the aforementioned mean scaled absolute error  $MSAE_{\sigma}$ . The second metric is the mean Intersection over Union (IoU) metric calculated over the segmentation image subset introduced in [13] containing 14 pixel-wise annotated images

Table 1: Comparison of training with weekly images with only the original labels and daily images with original and interpolated labels. Abbreviations used:  $\text{MSAE}_\sigma$  - Mean Scaled Absolute Error, IoU - Intersection over Union, DPC - DCA-Procrustes-Correlation. Top results are marked in **bold font**.

| Resolution Epochs |    | Weekly Images    |              |              | Daily Images     |              |              |
|-------------------|----|------------------|--------------|--------------|------------------|--------------|--------------|
|                   |    | MSAE $_{\sigma}$ | IoU          | DPC          | MSAE $_{\sigma}$ | IoU          | DPC          |
| 1536x768          | 3  | 0.527            | 0.158        | 0.724        | 0.499            | 0.198        | 0.780        |
|                   | 6  | 0.505            | 0.192        | 0.766        | 0.502            | 0.204        | 0.766        |
|                   | 10 | 0.501            | 0.196        | <b>0.770</b> | 0.503            | 0.199        | 0.772        |
|                   | 15 | 0.500            | 0.201        | 0.768        | 0.498            | 0.194        | 0.773        |
|                   | 25 | 0.501            | 0.203        | 0.760        | -                | -            | -            |
|                   | 40 | 0.502            | 0.187        | 0.765        | -                | -            | -            |
| 2688x1536         | 3  | 0.545            | 0.156        | 0.656        | 0.494            | 0.205        | 0.780        |
|                   | 6  | 0.510            | 0.188        | 0.757        | 0.493            | <b>0.223</b> | 0.778        |
|                   | 10 | 0.502            | 0.204        | 0.766        | 0.491            | 0.208        | 0.777        |
|                   | 15 | 0.497            | <b>0.208</b> | 0.757        | <b>0.489</b>     | 0.181        | <b>0.781</b> |
|                   | 25 | <b>0.493</b>     | 0.205        | 0.763        | -                | -            | -            |
|                   | 40 | 0.495            | 0.192        | 0.761        | -                | -            | -            |

from the InsectArmageddon dataset. The last metric we will refer to as the DCA-Procrustes-Correlation (DPC). It is calculated by performing a Detrended Correspondence Analysis (DCA) [10] on the target and predicted outputs, which are then compared with a Procrustes analysis. This returns a correlation value, where higher values show a higher similarity of the distributions to each other, which is significant for ecological applications.

With the  $\text{MSAE}_\sigma$ , we can evaluate the performance of our models in absolute terms, i.e., how accurate the species-wise predictions are based on the reference estimates. The IoU determines how well the top layer of plants is predicted, disregarding any occluded plants, and the DPC explains how well the predicted species distribution matches with the one estimated by the expert. All experiments are performed in a 12-fold cross-validation over three repetitions.

#### 4.4 Label Interpolation

The results of our experiments with label interpolation are shown in Table 1. Regarding the weekly annotated images, it is visible that the  $\text{MSAE}_\sigma$  and IoU are increasing with higher epoch counts, and the higher-resolution images also return slightly better results than the low-resolution images. However, after ten epochs, low-resolution images achieve the best DPC value (0.77). This value is not outperformed when using high-resolution images, leading to the conclusion that the network can learn to reduce the prediction error for some more dominant species from the high-resolution images but cannot accurately learn and reflect the actual distribution due to the neglect of less abundant species.



When looking at the results of the experiments using the interpolated daily images in conjunction with the weekly annotations, we notice improvements for both image resolutions, showing that our interpolation method is effective and leads to better results than just using the annotated images. The low-resolution and high-resolution images with daily images outperform their counterparts in all metrics. It should also be noted that the top performance is achieved after a smaller number of epochs for the daily images, likely because the number of iterations per epoch is about seven times the one with weekly images. This way, the top DPC value is achieved after three epochs, while the top IoU is achieved after six epochs. The  $\text{MSAE}_\sigma$  appears to be still improving for a larger number of epochs.

#### 4.5 Monte-Carlo Cropping

The results of our experiments with MCC on full-resolution images with weekly labels and interpolated daily labels are shown in Figure 3. The detailed numerical results can be found in the Supplementary Material. Generally, we can see that larger patch sizes lead to better results in terms of  $\text{MSAE}_\sigma$  and DPC, with the patch sizes 512 and 1024 yielding the top results for the experiments with daily and weekly labels. The patch size of 512 yields the best results for DPC, while the size of 1024 performs best in terms of  $\text{MSAE}_\sigma$ . For the weekly labels, the top DPC value and  $\text{MSAE}_\sigma$  value are 0.777 and 0.489, which outperform the top results on the full-resolution weekly images with a DPC of 0.766 and  $\text{MSAE}_\sigma$  of 0.493, respectively. For achieving this performance, for the patch size 512 the optimal sample size is 8, and for 1024 it is 2, representing about 50% of the number of pixels of the original image. The same configurations generate the best  $\text{MSAE}_\sigma$  and DPC results for the daily images, with 0.487 and 0.784, respectively. The MC training outperforms the full image training also here in terms of  $\text{MSAE}_\sigma$  (0.487 vs. 0.489) and DPC (0.784 vs. 0.781).

Interestingly, the best top layer prediction results, i.e., segmentation results, were achieved with a much smaller patch size, of  $128^2$  pixels. For the weekly images and a sample size of 128, the top IoU was 0.220, again outperforming the naive full-resolution approach with an IoU of 0.208. Similarly, with a patch size of 128 and a sample size of 128, the training on the daily images yields an IoU outperforming the full-resolution training (0.232 vs. 0.223).

The discrepancies between the configurations for optimal IoU and optimal  $\text{MSAE}_\sigma$  and DPC can be explained by what kind of features the network learns for the different patch sizes. With smaller patches, the network is forced to focus on the single plants shown in the top layer and learns little about the relationships of plants between each other, like occlusion. These relationships, however, play a significant role in the accurate prediction of the species-wise values and the entire composition, which are evaluated by  $\text{MSAE}_\sigma$  and DPC, respectively. Larger patch sizes capture the relationships and therefore perform better in these aspects.

In summary, our MCC approach can outperform the training with full-resolution images. The results significantly depend on the patch size, with higher

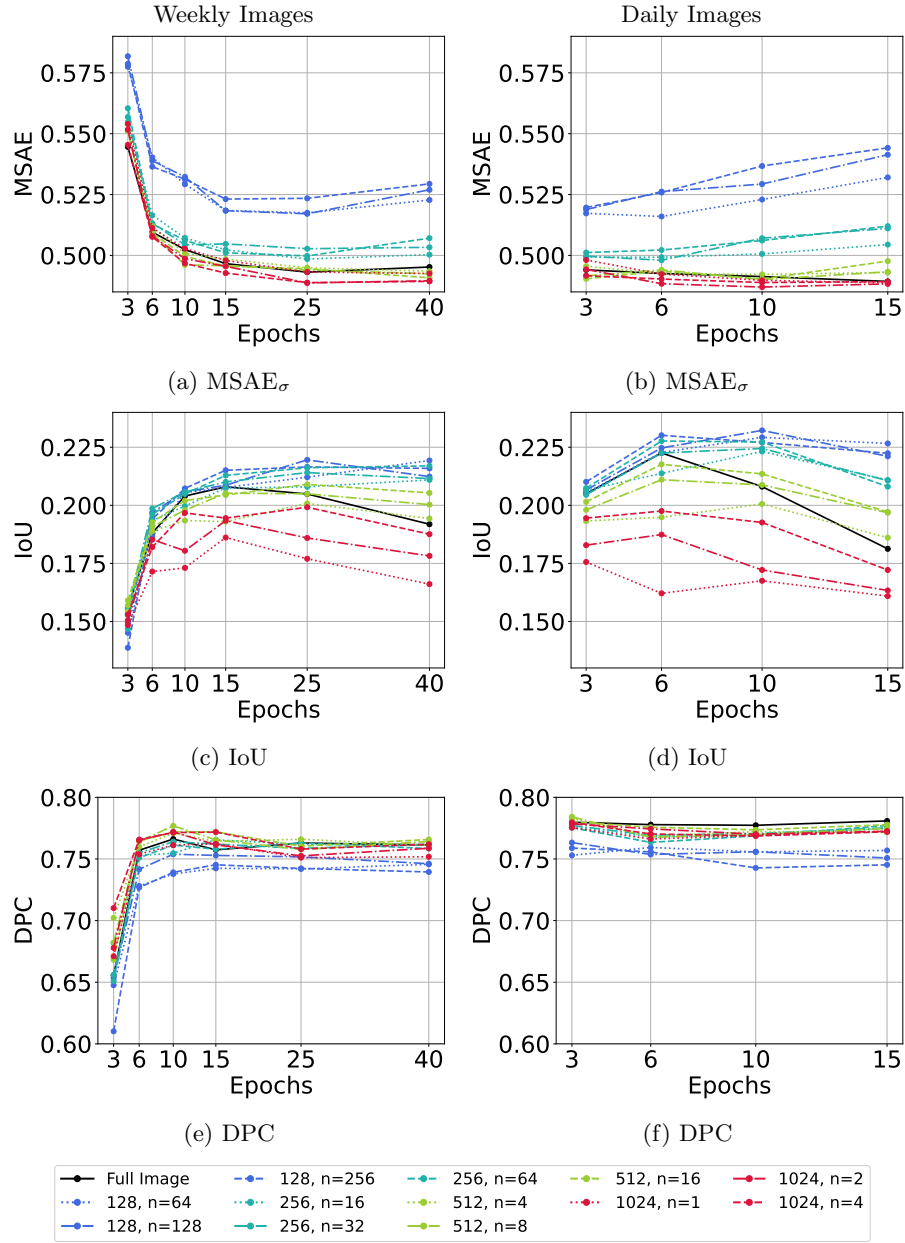


Fig. 3: The development of the different metrics over several training durations for images with weekly labels (left) and images with interpolated (daily) labels (right). Abbreviations used:  $MSAE_{\sigma}$  - Mean Scaled Absolute Error, IoU - Intersection over Union, DPC - DCA-Procrustes-Correlation

Table 2: Comparison of training speed per epoch on the full resolution images using different patches and sample sizes. Times shown are in minutes:seconds.

| Patch Size #Patches |     | Weekly Images<br>Time per Epoch | Daily Images<br>Time per Epoch |
|---------------------|-----|---------------------------------|--------------------------------|
| -                   | -   | 01:51                           | 16:17                          |
| 128                 | 64  | 00:55                           | 09:06                          |
|                     | 128 | 01:34                           | 14:01                          |
|                     | 256 | 03:01                           | 24:25                          |
| 256                 | 16  | 00:46                           | 08:22                          |
|                     | 32  | 01:08                           | 10:37                          |
|                     | 64  | 02:10                           | 18:22                          |
| 512                 | 4   | 00:46                           | 08:07                          |
|                     | 8   | 01:03                           | 10:16                          |
|                     | 16  | 01:55                           | 16:41                          |
| 1024                | 1   | 00:45                           | 08:15                          |
|                     | 2   | 01:02                           | 10:23                          |
|                     | 4   | 01:52                           | 16:17                          |

patch sizes resulting in better community-based predictions and smaller ones in better individual-based predictions. As training on smaller patches instead of a large image has computational implications, we will compare computation times in the following.

*Computation Time Comparison* A comparison of the times per epoch for each setup using the full-resolution images is shown in Table 2. These measurements were taken when training on about 75% of the images of the weekly and daily image sets on an RTX 3090. The training on the original full-resolution images took 1 minute and 51 seconds per epoch for only the weekly images and 16 minutes and 17 seconds for daily images. The top results for  $\text{MSAE}_\sigma$  and DPC were generated by patch size 512 and 8 patches sampled. This setup takes 1 minute and 3 seconds on the weekly images and 10 minutes and 16 seconds on daily ones, resulting in a time reduction of about a third. As the setup with patch size 512 and 4 sampled patches performs comparably well, one could even reduce the training time further by about 50%, at little cost in performance. The setup generating the top segmentation results, i.e., patch size 128 and sample sizes 256 and 128, differ in the training durations. Considering the larger numbers of pixels used for sample size 256, the duration is longer for this setup, requiring about 63% more time for an epoch. In contrast, the sample size of 128 reduces the training time again by about 14%. It should be noted that, with MCC, the number of epochs required for the optimal results on the weekly images are similar to the number of epochs for full-resolution training, and on the daily images the training times are usually even shorter for MCC training. For example, with MCC training on daily images the best DPC value is achieved after 3 epochs

as opposed to 15 for full-resolution training. Hence, the training times are not only reduced regarding the per-epoch duration, but also the number of epochs in total.

## 5 Conclusion & Future Work

We introduced two approaches for improving the data efficiency for plant cover estimation training. One method utilizes the unannotated images in the dataset, which can be collected at almost no cost; the other one enables efficient training at high resolution, gathering more information from the high resolution of the images.

Both approaches have proven effective: the label interpolation led to sufficient training data to receive improved results for all investigated metrics when using full-resolution images compared to their lower-resolution counterparts. Therefore, it is advantageous to collect more images than can be labeled, if the images are similar enough to existing images, as we can artificially increase the size of the dataset by interpolating. Furthermore, the Monte-Carlo Cropping improved these results even further, producing better results for different aspects of the plant cover prediction task while decreasing the training time and computation required during training. While, of course, a higher image resolution contains more information, with our MCC approach, especially in combination with label interpolation, such a high resolution can be utilized much more effectively. By increasing the image resolution in future experiments even further than in the InsectArmageddon experiment, with our methods such a high resolution can actually be utilized to improve the cover estimates without additional human effort.

Our approaches will be evaluated further on new plant cover datasets with different image resolutions, more frequent images and more plant species in future work. Moreover, the label interpolation might be improved with a more sophisticated interpolation model instead of linear interpolation, for example, a model that considers the information in the image for interpolation. Similarly, the sample selection of the Monte-Carlo Cropping could be improved by intelligently selecting the patches that contain the most information in the future.

## Acknowledgements

Matthias Körschens thanks the Carl Zeiss Foundation for the financial support. We thank Alban Gebler for enabling the image collection process in the iDiv EcoTron. We acknowledge funding from the German Research Foundation (DFG) via the German Centre for Integrative Biodiversity research (iDiv) Halle-Jena-Leipzig (FZT 118) for the support of the FlexPool project PhenEye (09159751).

## References

1. Andrzejak, M., Korell, L., Auge, H., Knight, T.M.: Effects of climate change and pollen supplementation on the reproductive success of two grassland plant species. *Ecology and Evolution* **12**(1), e8501 (2022)
2. Bolzano, B.: *Beyträge zu einer begründeteren Darstellung der Mathematik*, vol. 1. Im Verlage bey Caspar Widdmann (1810)
3. Bruehlheide, H., Dengler, J., Purschke, O., Lenoir, J., Jiménez-Alfaro, B., Hennekens, S.M., Botta-Dukát, Z., Chytrý, M., Field, R., Jansen, F., et al.: Global trait–environment relationships of plant communities. *Nature Ecology & Evolution* **2**(12), 1906–1917 (2018)
4. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 12475–12485 (2020)
5. Cui, Y., Song, Y., Sun, C., Howard, A., Belongie, S.: Large scale fine-grained categorization and domain-specific transfer learning. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4109–4118 (2018)
6. Evans, M.J., Rosenthal, J.S.: *Probability and statistics: The science of uncertainty*. Macmillan (2004)
7. Gerstner, K., Dormann, C.F., Stein, A., Manceur, A.M., Seppelt, R.: Editor’s choice: Review: Effects of land use on plant diversity—a global meta-analysis. *Journal of Applied Ecology* **51**(6), 1690–1700 (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 770–778 (2016)
9. Helm, J., Dutoit, T., Saatkamp, A., Bucher, S.F., Leiterer, M., Römermann, C.: Recovery of mediterranean steppe vegetation after cultivation: Legacy effects on plant composition, soil properties and functional traits. *Applied Vegetation Science* **22**(1), 71–84 (2019)
10. Hill, M.O., Gauch, H.G.: Detrended correspondence analysis: an improved ordination technique. In: *Classification and ordination*, pp. 47–58. Springer (1980)
11. Kahn, J., Lee, A., Hannun, A.: Self-training for end-to-end speech recognition. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 7084–7088. IEEE (2020)
12. Körschens, M., Bodesheim, P., Denzler, J.: Beyond global average pooling: Alternative feature aggregations for weakly supervised localization. In: *VISIGRAPP* (2022)
13. Körschens, M., Bodesheim, P., Römermann, C., Bucher, S.F., Migliavacca, M., Ulrich, J., Denzler, J.: Weakly supervised segmentation pretraining for plant cover prediction. In: *DAGM German Conference on Pattern Recognition*. pp. 589–603. Springer (2021)
14. Körschens, M., Bodesheim, P., Römermann, C., Bucher, S.F., Ulrich, J., Denzler, J.: Towards confirmable automated plant cover determination. In: *ECCV Workshop on Computer Vision Problems in Plant Phenotyping (CVPPP)* (2020)
15. Li, C., Li, L., Geng, Y., Jiang, H., Cheng, M., Zhang, B., Ke, Z., Xu, X., Chu, X.: Yolov6 v3.0: A full-scale reloading. *arXiv preprint arXiv:2301.05586* (2023)
16. Li, S., Wang, Z., Liu, Z., Tan, C., Lin, H., Wu, D., Chen, Z., Zheng, J., Li, S.Z.: Efficient multi-order gated aggregation network. *arXiv preprint arXiv:2211.03295* (2022)

17. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2117–2125 (2017)
18. Liu, H., Mi, Z., Lin, L., Wang, Y., Zhang, Z., Zhang, F., Wang, H., Liu, L., Zhu, B., Cao, G., et al.: Shifting plant species composition in response to climate change stabilizes grassland primary production. *Proceedings of the National Academy of Sciences* **115**(16), 4051–4056 (2018)
19. Lloret, F., Peñuelas, J., Prieto, P., Llorens, L., Estiarte, M.: Plant community changes induced by experimental climate change: seedling and adult species composition. *Perspectives in Plant Ecology, Evolution and Systematics* **11**(1), 53–63 (2009)
20. Rosenzweig, C., Casassa, G., Karoly, D.J., et al.: Assessment of observed changes and responses in natural and managed systems. *Climate Change 2007: Impacts, Adaptation and Vulnerability. Contribution of Working Group II to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change* pp. 79–131 (2007)
21. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* **115**(3), 211–252 (2015)
22. Scudder, H.: Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory* **11**(3), 363–371 (1965)
23. Souza, L., Zelikova, T.J., Sanders, N.J.: Bottom-up and top-down effects on plant communities: nutrients limit productivity, but insects determine diversity and composition. *Oikos* **125**(4), 566–575 (2016)
24. Ulrich, J., Bucher, S.F., Eisenhauer, N., Schmidt, A., Türke, M., Gebler, A., Barry, K., Lange, M., Römermann, C.: Invertebrate decline leads to shifts in plant species abundance and phenology. *Frontiers in plant science* **11**, 1410 (2020)
25. Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., et al.: Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv preprint arXiv:2211.05778* (2022)
26. Yuan, Y., Chen, X., Chen, X., Wang, J.: Segmentation transformer: Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065* (2019)
27. Zhang, H., Cissé, M., Dauphin, Y., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. *ArXiv abs/1710.09412* (2017)
28. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2921–2929 (2016)