# Gradient-based Active Learning for Semantic Segmentation of Crop and Weed for Agricultural Robots

Rasha Sheikh Philipp Lottes Andres Milioto Cyrill Stachniss Maren Bennewitz Thomas Schultz

***Abstract*— Annotated datasets are essential for supervised learning. However, annotating large datasets to train deep neural networks that perform well is a tedious and time-intensive task. This paper addresses active learning in the context of semantic segmentation with the goal of reducing the human labeling effort. Our target application is agricultural robotics and we focus on the task of distinguishing between crop and weed plants from image data. A key challenge in this application is the transfer of an existing semantic segmentation CNN to a new field. We propose a novel a approach that, given a trained model on one field, refines the network on a substantially different field providing an effective method of selecting samples to annotate for supporting the transfer. Our method takes into account the influence of the unlabeled samples on the weights of the network and ranks and selects them accordingly for annotation. We evaluated our approach on two challenging datasets from the agricultural robotics domain and show that we achieve a higher accuracy with a smaller number of samples compared to random sampling for annotation as well as uncertainty-based approaches to select examples for annotation. Thus, our approach reduces the required human labeling effort.**

## I. INTRODUCTION

The ability to semantically interpret the scene in front of a robot is key for intelligent behavior in several applications. For example, precision farming robots need to know which type of plant they perceive or autonomous cars need to know which object in their surroundings is a car, a pedestrian, or a cyclist. These classification or semantic segmentation tasks are typically tackled using convolutional neural networks (CNNs) operating on image data. In order to perform well, neural networks need to be trained with appropriately annotated datasets.

The performance of most supervised learning approaches and especially deep learning systems is related to the quality and quantity of training data. Annotated training data, however, has a high cost as often a larger number of labeled training data is required. In this work, we focus on optimizing the training set generation for semantic segmentation of image data obtained from a mobile robot. Semantic segmentation refers to the task of computing a pixel-wise labeling of the images. More concretely, we address the agricultural robotics application in which robots should perform automated weed control. For the semantic segmentation, this means that we need to compute the semantic label "crop", "weed", or "misc" for every pixel in the image. This task is particularly challenging as the field conditions often change substantially between years, regions, weather, and soil conditions. Thus, one often tries to adapt and refine existing semantic segmentation systems to new field conditions through annotated data from the new field. As these new annotation need to be executed at the end-users site, one is interested in keeping this effort as low as possible. Thus, this problem is a perfect domain for active learning approaches trying to reduce the required amount of data to be annotated.

Given annotated data on one agricultural field and a network that was trained on it, we address the problem of transferring this knowledge to new fields with minimum effort. Datasets from different fields reveal different crop and weed statistics. They almost always differ by soil type, weather condition, or various small objects that can be found on the ground, such as stones, dried vegetation, or marks from agricultural machines, i.e. patterns that are neither crop nor weed. Additionally, the robot can acquire images of plants at a certain growth stage in one field, while the growth state on the target field is different. Lastly, artifacts such as contrast changes can be found in the camera images captured from the various locations. These conditions make it difficult to simply reuse a previously trained network from one field and infer the labels on another, see also [10], [11]. Thus, the network has to be re-trained on annotated images taken in the new field and we propose an active learning approach to select samples that the network will most benefit from and will generalize to the rest of the unlabeled data while minimizing the effort of annotating images.

The main contribution of this work is an active learning approach that intelligently picks images taken under the new conditions based on the effect these training samples will have on the weight gradients of the CNN. Our strategy is based on the observation that given a trained model and unseen samples from a different domain, the samples that the network performs most poorly on, especially at the beginning, will have the largest weight gradients and consequently the largest impact on the weights. Our approach selects samples in batches, each time refining the network, then computing a new ranking of the unlabeled data. The best samples are then selected and the network is refined a gain. To compute the gradients, corresponding ground truth data is needed, which we assume does not exist. We circumvent this by using pseudo ground truth that we obtain with unsupervised segmentation. We implemented and evaluated framework on agricultural datasets [3] with different characteristics. Our results indicate that our method produces a higher accuracy on both datasets with a fewer number of samples compared to random sampling for annotation as well as uncertainty based approaches.

All authors are with the University of Bonn, Germany. This work has partly been supported by the DFG-funded Cluster of Excellence EXC 2070 PhenoRob.

Manuscript 1089 submitted to 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2019.

## II. RELATED WORK

There have been numerous works on general active learning in the past [16], [7], [8]. Recently, the research topic of using active learning in combination with deep learning has received attention. We focus in this section on the different approaches that explored active learning within a deep learning framework.

In [20], the authors define measures of entropy and diversity to select new samples for annotation. The entropy of a patch is calculated based on the classification uncertainty of the network, whereas the diversity is computed using the Kullback Leibler divergence of different patches within the same sample candidate. A pretrained network is then refined using the samples with the highest entropy and diversity.

The authors of [18] select samples that the network is uncertain of and that are representative of other images in the dataset. The uncertainty is measured by bootstrapping, where multiple fully convolutional networks (FCNs) are trained, and the variance among these trained models is used to estimate the uncertainty. In order to choose samples that are highly similar to others in the training set, features are extracted from the encoding part of the network, and the cosine similarity between pairs of images is calculated.

In [4], foreground masks are create in an iterative manner. Samples that are deemed most valuable for annotation are selected. Their ground truth annotation is then propagated to new samples and the process is repeated. To pick samples for which human annotation will propagate well, the authors build a Markov Random Field (MRF) joint segmentation graph. The graph is then used to find samples that have the largest influence, diversity and uncertainty. The influence and diversity are computed using the cosine-similarity of different images features, while the uncertainty is estimated using a regressor that quantifies the quality of a prediction.

Different acquisition functions were evaluated in [6]. An active learning system would use such an acquisition function to choose the best next sample to annotate. These functions include maximizing the predictive entropy of a model given the training set and a new sample, and closely related to that is the variation ratio measure. Another function maximizes the mutual information between predictions and the model posterior. These different measures are compared using a Bayesian Convolutional Neural Network that has a prior probability distribution over the model parameters.

Uncertainty estimation for active learning can be performed using Monte-Carlo dropout as in [6] or with an ensemble of deep networks. These uncertainties can then be used in the different acquisition functions described earlier. The authors of [2] compare both of these approaches on different datasets. They found that an ensemble of deep classifiers has a superior performance even with a smaller number of models. They conclude that Monte-Carlo dropout approaches suffer from a lower diversity and a smaller model capacity.

The authors of [15] assert based on the experiments they performed that uncertainty based approaches are not effective for active learning with CNNs. They hypothesize that this is not due to the inaccurate estimate of uncertainty by the network, rather by the ineffectiveness of uncertainty based approaches to cover the space of image features. They instead propose to choose samples such that the largest distance between a new sample and its closest neighbor in the selected subset is minimized.

In [17], two sets of samples are selected for annotation that can then be used by the network to refine the model. The first set consists of samples that the network is uncertain about. These include samples with the lowest softmax confidence values, samples with the highest entropy, and lastly samples with a small margin of probability difference between the most probable class and the second most probable class. This first set is then presented to a human for annotation. The second set consists of samples that the network is highly certain about, these are assigned their predicted classes as pseudo labels and added to the set of training samples without asking a human to annotate them.

The Expected Model Output Change Principle (EMOC) developed by [5] tries to avoid selecting samples that are redundant and [9] follow this approach with deep neural networks. This principle measures how a model would perform with and without the candidate sample. Given that the labels are unknown, a marginalization over the possible labels is needed. This marginalization however can be expensive when having a large number of classes, so the authors use maximum a-posteriori approximation instead and use the class with the highest probability prediction.

In the context of self-learning, the authors of [19] use labels obtained with K-means graph cuts as ground truth for their network. The predictions produced by the model are then used as the target labels for the next iteration of the process.

Different to these works, we experiment with approaches that directly measure how annotated samples can affect the gradients. We use labels obtained with unsupervised segmentation as pseudo ground truth and compute the gradients w.r.t the weights. We then refine a pre-trained network with the newly annotated samples in an iterative manner.

## III. SEGMENTATION FRAMEWORK

We use Bonnet [13] to train a model on the Bonn dataset. We then refine the trained model on other datasets by incrementally selecting batches of samples. The datasets differ in their crop/weed statistics and the images acquired with the cameras also differ in their illumination. Therefore, simply running the trained model to segment the vegetation in other fields does not work. We briefly first describe the architecture of the network used, then present different methods to select samples for refinement that we experimented with.

Bonnet is an open-source deep network framework developed by [13]. It was designed with efficiency in mind so that it is able to run at 20 Hz. The network is based on SegNet [1] and ENet [14]. It has an encoder-decoder structure with a total of 25 [5x5] convolutional layers. It uses batch normalization, residual connections, and ReLU as the non-linearity layer.

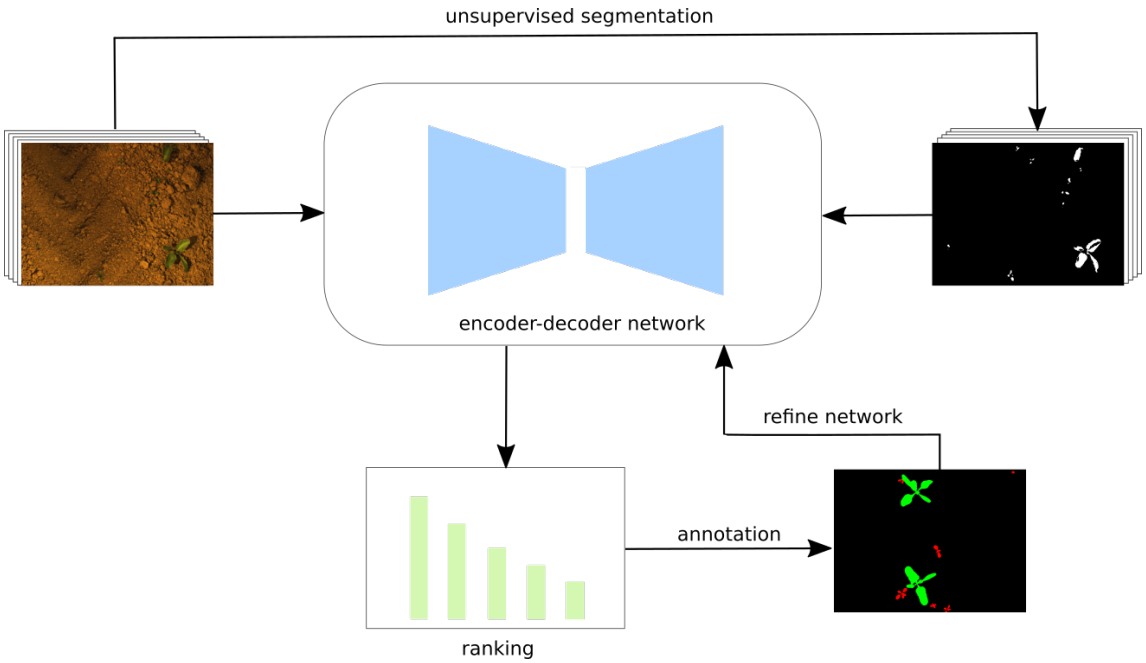



Fig. 1. Overview figure of our system. We first perform unsupervised segmentation to obtain pseudo ground truth. Given the labels and different measures produced by the network, we rank the unlabeled samples and pick them accordingly for annotation. These are then used to refine the network.

To speed up prediction, the authors replace the [5x5] conventional convolutional layer with a mix of [1x1] convolutions and separable [1x5] and [5x1] convolutions. Additionally, instead of using the relatively more expensive transposed convolutions in the decoder, unpooling is done using the respective pooling indices in the encoder part.

As input to our network we only use the RGB channels.

## IV. Effective Sample Selection

We experimented with different approaches to select samples. The baseline is randomly selecting samples for annotation in batches of 10. The other approaches include selecting samples driven by the uncertainty of the network, the training loss, and the weights gradients of the network.

### A. Uncertainty

To infer the pixel-wise semantic segmentation of a new image, the network computes softmax probabilities in its last layer. The probabilities can serve as a guide as to which samples the network is most uncertain of. For every image passed through the network, we compute the following measure of the prediction confidence:

$$u(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \max_{c} p(c|x_i), \tag{1}$$

where $x_i$ is pixel $i$ in the image and $c$ is the predicted class.

We then sort the images based on the computed uncertainty measure and pick the images accordingly to refine the network on a new dataset. The images are selected on a log-space scale, rather than selecting those with the highest uncertainty, as we found out that the network learns better when presented with diverse samples. The log-space approach is used in the following methods as well.

### B. Loss

The loss of the network is an indication of the segmentation error and given that the gradients w.r.t the network weights are driven by the loss, it provides a useful cue as to which samples the network will most benefit from. We first use unsupervised segmentation to produce pseudo-ground truth for the images. We then run the network on all the images and compute the loss with these labels as ground truth. The images are sorted based on this loss and again chosen on a log-space scale. We note that the pseudo-ground truth is only used to compute the loss but the network weights remain unchanged. They are only later updated with the manual annotations of the selected samples.

For unsupervised segmentation, we only do foreground-background segmentation. In the train phase, we run k-means clustering on the RGB channels of 10 random samples of the new dataset to learn 20 clusters. The annotator can then look at a single image and choose those clusters that separate vegetation from soil. In our experiments, two clusters were enough. The rest of the images can then be annotated using those clusters.

### C. Norm of Gradients

For this approach and the following one, we pick those samples for annotation that might have the largest impact on the network weights. The norm of the network gradients is a measure that is indicative of which samples will affect the weights more than others. As in the previous approach, we use labels from unsupervised segmentation as pseudo-ground

truth. We run the network on the training images for one epoch and compute the gradients. Again we note that this step is only used to compute the gradients but we don't change the network weights. Once we have the gradients, we compute the $L_2$ norm of those in the last layer of the network (the classifier layer).

$$n_g(\mathbf{x}) = \left\| \nabla_{w_f} \mathcal{L}(\mathbf{x}) \right\|, \tag{2}$$

where $w_f$ are the weights of the final layer.

The images are sorted based on this measure and we pick samples on a log-space scale afterwards.

### D. Gradient Projection

The log-space in the previous approaches was used to ensure there is enough diversity among the samples so that the network does not overfit on them and can generalize to unseen data. Here we use a different method that relies on the space spanned by the gradients where we project onto the orthogonal complement of the gradients of the selected samples. For every sample picked we project the gradients of all remaining samples onto the selected sample gradient. We then subtract the projected gradient from the original gradients. The residual we are left with indicates which samples have the strongest remaining effect on the weights after accounting for the already selected samples. This can be formulated as:

$$n_p(\mathbf{x}) = \left\| \mathbf{g_x} - \sum_{i=1}^{S} \frac{\langle \mathbf{g_s}, \mathbf{g_x} \rangle}{\langle \mathbf{g_s}, \mathbf{g_s} \rangle} \mathbf{g_s} \right\|, \tag{3}$$

where $g_s$ is the gradient of the previously selected sample, and $g_x$ is the gradient of the current sample.

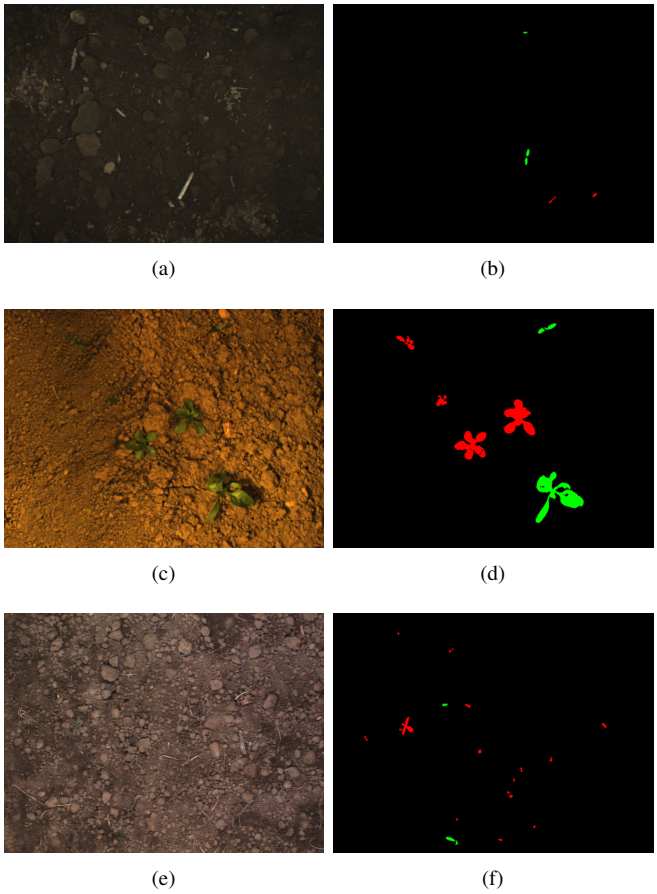
We select samples one by one, each time sorting them according to this measure and choosing the one with the highest norm of the residual. To pick the first sample, we choose that with the highest norm of the gradient.

## V. EXPERIMENTAL EVALUATION

We show in this section the effectiveness of the approaches we designed for active learning, where samples are selected using the different methods and the performance is tested on different datasets.

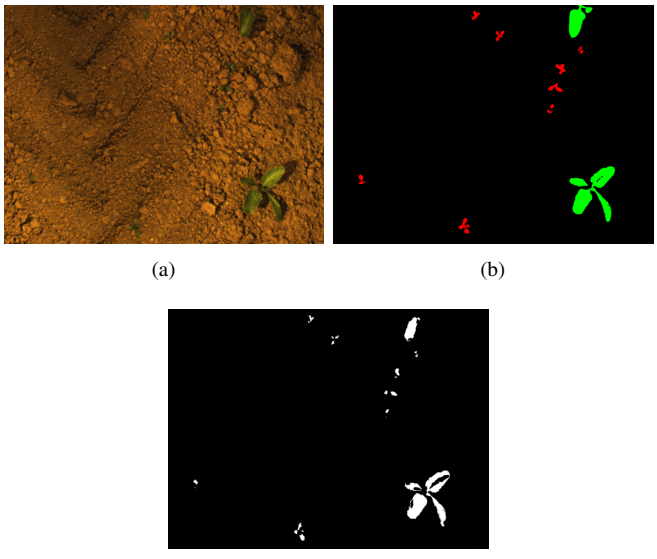
### A. Datasets

The datasets we used were acquired with a Bosch Deepfield Robotics BoniRob UGV in three different fields: Bonn and Stuttgart in Germany, and Zurich in Switzerland. The datasets have weed and crop plants at different stages of growth. Figure 2 shows sample images from the different datasets. The images vary in their illumination, soil type, and classes statistics, hence the need for transfer learning. The images have been annotated into three classes: soil, weed and crop.The Bonn dataset is partly publicly available [3]. Table I shows the number of images in each dataset and the ratio of foreground pixels. It can be clearly seen that there is a high imbalance of classes in the data.



(a) (b) (c) (d) (e) (f)

Fig. 2. Sample images from the Bonn, Stuttgart, and Zurich datasets in the first, second, and third row respectively. The first column shows the RGB images and the second column shows their annotations. Green denotes crop while red denoted weed.



(a) (b) (c)

Fig. 3. Foreground segmentation of vegetation. Note that only a rough segmentation is enough for our approach.

TABLE I

DATASETS STATISTICS OF CROP AND WEED PLANTS

| | Bonn | Stuttgart | Zurich |
|---|---|---|---|
| Images | 8230 | 2584 | 2577 |
| Crop pixels | 2.0% | 1.5% | 0.4% |
| Weed pixels | 0.3% | 0.7% | 0.1% |

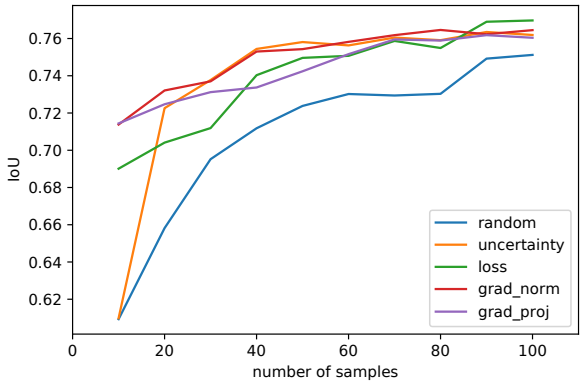



Fig. 4. Pixel-wise mean IoU on the Stuttgart dataset.

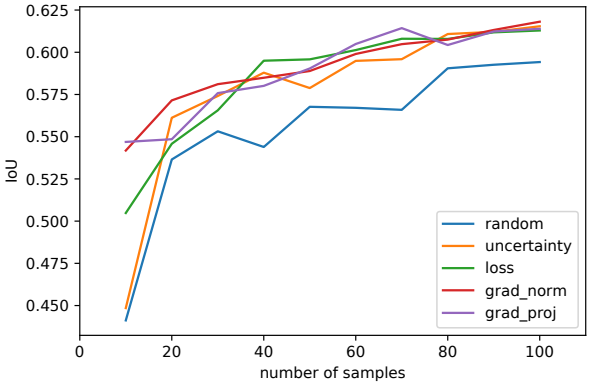



Fig. 5. Pixel-wise mean IoU on the Zurich dataset.

### B. Experiment Setup

We evaluate our different approaches by first training a network on the Bonn dataset then refining it on the Stuttgart and Zurich datasets. To refine the network we pick unlabeled samples in batches of 10 using one of the methods described in Section III. Once they are annotated, they are presented to the network. We repeat this process iteratively, each time refining the network on all of the newly annotated samples.

For the methods presented in sections IV-B, IV-C, and IV-D, we first obtain foreground masks with unsupervised segmentation. Figure 3 shows an image, its ground truth and the foreground segmentation provided by clustering. Note that a rough segmentation is enough to be useful for these approaches.

We follow the approach of [12] and split the new dataset into three sets: 40% for training, 10% for validation, and 50% for testing. The samples are picked from the training set. All experiments were conducted on four Titan X GPUs.

### C. Results

Figures 4 and 5 show the pixel-wise mean intersection over union (mIoU) on the Stuttgart and Zurich datasets when selecting samples for annotation with different methods. It can be seen from the plots that methods that take into account the impact of the samples on the weights lead to better generalization to the rest of the unseen data, even when presented with a small number of annotated images. In particular, ranking the samples based on the norm of the gradients results in higher mIoU on both datasets.

To further quantify the performance of our approach, we use the object-wise metric defined by [12], where the accuracy is measured for objects larger than 50 pixels. Since the target application is weeding with agricultural robotics, this metric is more directly useful than pixel-wise performance.

Tables IV and V show how our approach performs on the Stuttgart and Zurich datasets. Each row shows the mean accuracy when selecting $n$ samples with different methods. The baseline is random sampling shown in the first column.

A few observations can be made: the effect of the sampling method is more pronounced when only a few images are selected. Again, it can be seen that methods measuring the influence of the samples on the weighs perform better. For instance, training a model with 20 samples picked with the gradient norm method produces accuracies that can only be achieved when picking 60 samples with the random method, lowering the annotation effort considerably.

As the model is trained on more and more samples, the accuracy plateaus as is expected and the variation between the different methods decreases. It can be noted however that random sampling has a lower performance even with a greater number of images, and even produces fluctuating results on the more challenging Zurich dataset.

The gradient norm method shows consistent improvement over other methods for different number of samples and across the two datasets, confirming our intuition that samples that might have a larger influence on the weights are more valuable for annotation, as the network can benefit more from them.

Projecting out the gradients iteratively seems to be a promising method. For small number of images, it performs best but then the improvement afterwards is not as noticeable. Investigating the images that are being chosen in the fourth row and beyond for this method, we hypothesize that although these images will force the network weights to change in orthogonal directions, they might not be representative of a larger subset of images in the dataset. We plan to investigate this further in the future.

A more detailed breakdown of the methods performance is shown in tables II and III. Table II shows the pixel-wise precision and recall on the Stuttgart dataset after selecting the first 10 samples. Both methods, Gradient Norm and

TABLE II

PIXEL-WISE PRECISION AND RECALL ON THE STUTTGART DATASET AFTER SELECTING THE FIRST 10 SAMPLES

| | Precision | | Recall | |
|---|---|---|---|---|
| | Weed | Crop | Weed | Crop |
| Random | *0.4095* | 0.7278 | 0.4851 | *0.6946* |
| Uncertainty | 0.5580 | *0.6646* | *0.2711* | **0.8880** |
| Loss | 0.5331 | 0.8025 | 0.6179 | 0.8112 |
| Gradient Norm | **0.5970** | 0.8259 | 0.6136 | 0.8402 |
| Gradient Projection | 0.5745 | **0.8365** | **0.6564** | 0.8212 |

TABLE III

OBJECT-WISE PRECISION AND RECALL ON THE STUTTGART DATASET AFTER SELECTING THE FIRST 10 SAMPLES

| | Precision | | Recall | |
|---|---|---|---|---|
| | Weed | Crop | Weed | Crop |
| Random | *0.8723* | 0.5740 | 0.6587 | 0.6474 |
| Uncertainty | **0.9476** | *0.4586* | *0.4919* | **0.8854** |
| Loss | 0.9005 | 0.6898 | 0.7811 | 0.7351 |
| Gradient Norm | 0.9090 | **0.7390** | 0.7970 | 0.7536 |
| Gradient Projection | 0.9030 | 0.7308 | **0.8289** | 0.7375 |

Gradient Projection have a high recall and precision of the crop class without degrading those of the weed class. The object-wise performance in Table III further illustrates the effectiveness of these methods. Gradient Norm and Gradient Projection produce high precision and recall for both classes. The Uncertainty-based method, on the other hand, shows a large imbalance of performance on the two classes.

To analyze the samples picked by the gradient based approach, we plot the t-distributed Stochastic Neighbor Embedding (t-SNE) of the gradients in Figure 6. Each circle in the plot denotes the 2-D embedding of the gradients of a single image before picking the first 10 samples. Red circles in the figure denote images selected by the Gradient Norm method. Points in the bottom left have a large norm of gradients, while the point in the upper left has the lowest norm of the gradient. Given the log-spacing of the selected samples, it can be seen that it more densely selects images that the network finds difficult, and the other selected points are more spread out. For future work, we plan to investigate how this embedding can be further exploited to pick the best samples.

## VI. CONCLUSION

In this work, we design an approach for active learning that selects samples for annotation more effectively. We first obtain pseudo ground truth labels with unsupervised segmentation then use those labels to measure how the unlabeled samples might affect the weights if selected for training. We choose samples based on this ranking and refine the network on new datasets. We verified the performance of our gradient-based approach on two datasets that have different characteristics from the one which the network was pre-trained on. Our results show the effectiveness of our method as it produces higher accuracies with only a few number of samples, compared to random sampling or uncertainty based approaches. The effort in human annotation is thereby considerably reduced.

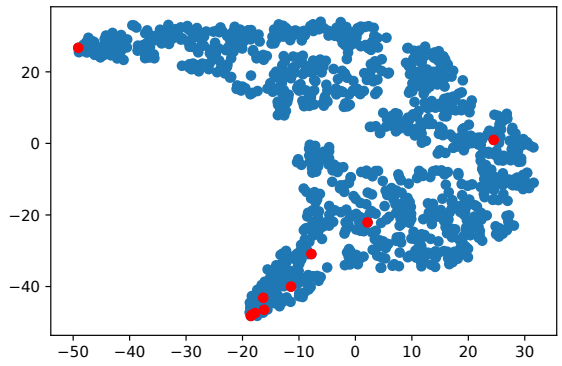



Fig. 6. t-SNE of the images gradients on the Stuttgart dataset. The first 10 samples selected by the gradient norm method are shown in red.

## REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[3] Nived Chebrolu, Philipp Lottes, Alexander Schaefer, Wera Winterhalter, Wolfram Burgard, and Cyrill Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 36(10):1045–1052, 2017.

[4] Suyog Dutt Jain and Kristen Grauman. Active image segmentation propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2864–2873, 2016.

[5] Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In *European Conference on Computer Vision*, pages 562–577. Springer, 2014.

[6] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

[7] Isabelle Guyon, Gavin Cawley, Gideon Dror, and Vincent Lemaire. Results of the active learning challenge. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pages 19–45, 2011.

[8] Alex Holub, Pietro Perona, and Michael C Burl. Entropy-based active learning for object recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[9] Christoph Käding, Erik Rodner, Alexander Freytag, and Joachim Denzler. Active and continuous exploration with deep neural networks and expected model output changes. *arXiv preprint arXiv:1612.06129*, 2016.

[10] P. Lottes, J. Behley, A. Milioto, and C. Stachniss. Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters (RA-L)*, 3:3097–3104, 2018.

TABLE IV

OBJECT-WISE PERFORMANCE ON THE STUTTGART DATASET. EACH ROW SHOWS THE PERFORMANCE AFTER SELECTING 10 SAMPLES WITH THE DIFFERENT METHODS AND REFINING THE NETWORK.

| Samples No. | Random | Uncertainty | Loss | Gradient Norm | Gradient Proj. |
|---|---|---|---|---|---|
| 10 | 0.6920 | 0.6437 | 0.7882 | 0.8040 | 0.8196 |
| 20 | 0.7402 | 0.8408 | 0.7769 | 0.8350 | 0.8404 |
| 30 | 0.8138 | 0.8359 | 0.7950 | 0.8461 | 0.8470 |
| 40 | 0.8254 | 0.8529 | 0.8555 | 0.8682 | 0.8252 |
| 50 | 0.8225 | 0.8529 | 0.8523 | 0.8599 | 0.8278 |
| 60 | 0.8308 | 0.8497 | 0.8596 | 0.8569 | 0.8384 |
| 70 | 0.8335 | 0.8542 | 0.8666 | 0.8622 | 0.8366 |
| 80 | 0.8321 | 0.8595 | 0.8455 | 0.8596 | 0.8386 |
| 90 | 0.8424 | 0.8565 | 0.8643 | 0.8639 | 0.8399 |
| 100 | 0.8394 | 0.8502 | 0.8638 | 0.8531 | 0.8529 |

TABLE V

OBJECT-WISE PERFORMANCE ON THE ZURICH DATASET. EACH ROW SHOWS THE PERFORMANCE AFTER SELECTING 10 SAMPLES WITH THE DIFFERENT METHODS AND REFINING THE NETWORK.

| Samples No. | Random | Uncertainty | Loss | Gradient Norm | Gradient Proj. |
|---|---|---|---|---|---|
| 10 | 0.7552 | 0.6943 | 0.7697 | 0.8354 | 0.8025 |
| 20 | 0.7971 | 0.8281 | 0.8189 | 0.8768 | 0.8170 |
| 30 | 0.8591 | 0.8674 | 0.8321 | 0.8553 | 0.8299 |
| 40 | 0.8575 | 0.8690 | 0.8610 | 0.8711 | 0.8479 |
| 50 | 0.8593 | 0.8547 | 0.8636 | 0.8852 | 0.8784 |
| 60 | 0.8666 | 0.8737 | 0.8805 | 0.8827 | 0.8895 |
| 70 | 0.8601 | 0.8664 | 0.8880 | 0.8827 | 0.8878 |
| 80 | 0.8241 | 0.8869 | 0.8867 | 0.8897 | 0.8784 |
| 90 | 0.8476 | 0.8667 | 0.8812 | 0.8928 | 0.8871 |
| 100 | 0.7911 | 0.8700 | 0.8873 | 0.8873 | 0.8805 |


[11] P. Lottes and C. Stachniss. Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[12] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2229–2235. IEEE, 2018.

[13] Andres Milioto and Cyrill Stachniss. Bonnet: An open-source training and deployment framework for semantic segmentation in robotics using cnns. *arXiv preprint arXiv:1802.08960*, 2018.

[14] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[15] Ozan Sener and Silvio Savarese. A geometric approach to active learning for convolutional neural networks. *arXiv preprint arXiv*, 1708:1, 2017.

[16] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[17] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2017.

[18] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 399–407. Springer, 2017.

[19] Ling Zhang, Vissagan Gopalakrishnan, Le Lu, Ronald M Summers, Joel Moss, and Jianhua Yao. Self-learning to detect and segment cysts in lung ct images without manual annotation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1100–1103. IEEE, 2018.

[20] Zongwei Zhou, Jae Shin, Lei Zhang, Suryakanth Gurudu, Michael Gotway, and Jianming Liang. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7340–7351, 2017.