Weed Detection in Close-range Imagery of Agricultural Fields using Neural Networks

AVISHEK DUTTA[1], JOSEPH M. GITAHI[1], PRASHANT GHIMIRE[1], ROBIN MINK[2], GERASSIMOS PETEINATOS[2], JOHANNES ENGELS[1], MICHAEL HAHN[1] & ROLAND GERHARDS[2]

Abstract: Modern day agriculture is becoming an endeavour where precision is highly desired and drones with imaging capabilities are contributors of big data to this field of research. Our focus in this paper is on precise weed control, in particular, the optimisation of yield and cost while having minimum impact on the environment. The information about in-field variability of weed patches can be exploited for sub-plot specific weed control, which leads to a restrained yet flexible use of herbicides. We use close-range imagery of weeds recorded with handheld cameras, having a resolution of only a few millimetres per pixel in their natural environment. In this paper we adapt Convolutional Neural Networks with the goal of separating weeds from the main crops in close-range imagery. We explore two ways to design the networks: pixel-wise classification and object-based detection. For both approaches, we use pre-trained networks, which are fine-tuned with the recorded weed images. The deep learning libraries used are Caffe and Tensorflow. The study demonstrates successful adaptation of pre-trained CNNs for weed classification in close-range imagery which could be extended to UAV imagery in future.

1 Introduction

Modern agricultural techniques focus on high yields, low costs and eco-friendly practices. While frugal use of herbicides is desired, weed control still remains important for the increase in the productivity of the crops. The common approach so far is the uniform application of herbicides to a field, neglecting the spatial variability of weed species and densities. This results in higher costs, pollution of soil and water resources. In addition, the herbicides may adversely affect the crops if applied in high concentrations. By mapping different species of weeds, their density and distribution, herbicide spraying can be adjusted as opposed to uniform application.

As weeds and crops are spectrally similar at an early stage of growth, pixel-based classifications of the two do not always provide reliable accuracy and hence popular rule-based methods include features such as the shape of leaf and texture to increase the achievable accuracy of classification (SØGAARD 2005; ÅSTRAND & BAERVELDT 2002; GOLZARIAN & FRICK 2011). However, these classification methods rely on handcrafted feature extraction techniques which are not robust enough to discern complexities that exist in the natural environment.

Our ultimate objective is to classify weeds in aerial images acquired from a height of less than 10 meters but for this study, we restrict ourselves to terrestrial images i.e. images acquired with hand-held devices. In this paper, we propose Convolutional Neural Networks (CNNs) for weeds classification using two approaches, object detection and pixel-wise labelling. We adapt pre-trained CNN models trained on expansive datasets and fine-tune with the weeds' images.

[1] Hochschule für Technik Stuttgart, Schellingstrasse 24, D-70174 Stuttgart, E-Mail: avishek.dutta@hft-stuttgart.de

[2] Universität Hohenheim, Institut für Phytomedizin, Otto-Sander-Str. 5, D-70599 Stuttgart, E-Mail: robin.mink@uni-hohenheim.de

2 Data

The data used for this study comprises images of five different weed species, as shown below.

Tab. 1: A list of the weeds classes available for the study

EPPO Code	Scientific name	English name
MATCH	Matricaria chamomilla	Wild chamomile
POAAN	Poa annua	Annual meadowgrass
STEME	Stellaria media	Common chickweed
VIOAR	Viola arvensis	Field pansy
AMARE	Amaranthus retroflexus	Common amaranth

The images were taken at Heidfeldhof farm, located in Plieningen, Stuttgart. These images were photographed vertically downwards from approximately 50cm above the ground by a cell phone camera having a resolution of 3024×4032 pixels. The weeds were cultivated in natural environment and were at an early stage of growth. Additionally some weed images from a dataset created by GISELSSON et al. (2017) were used for testing the models in the pixel-based approach. This dataset consists of images taken in controlled conditions with the soil covered by small stones to prevent green moss layer.

3 Object-based approach

Object detection deals with localization of objects in addition to the classification of the same objects. This approach of the study utilizes Faster R-CNN (REN et al. 2015) architecture for localizing weeds along with the recognition of the species to which they belong, by fine-tuning a pre-trained Resnet-50 model (HE et al. 2016) trained on 90 different object categories of COCO dataset (LIN et al. 2015).

Resnet-50 is a 50-layer architecture, consisting of the building blocks as shown in Table 2. A building block, for example, Conv2_x consists of three units with each unit having three convolutional layers of 1×1, 3×3 and 1×1. A shortcut connection is added from the result of the previous block (which in this example is 3×3 max pool) to the output of the first unit of Conv2_x. This shortcut connection is implemented as element-wise addition of the outputs of 3×3 max_pool to the output of the first sub-block of Conv2_x. Shortcut connections are also provided within the adjacent units in a block. For example, a shortcut connection is provided between the output of 1×1, 256 convolution of first unit of block Conv2_x and the output of 1×1, 256 convolution of the second unit of the same block. The shortcut connections between the units in adjacent block decrease the image size. Batch normalization has been performed after each convolution to overcome the problem of exploding and vanishing of the gradients during backpropagation.

Tab. 2: Resnet Architecture for an input size of 224*224 (adapted from HE et al. (2016))

Building block	Output size	Units
Conv1	112*112	7*7, 64, stride 2
	56*56	3*3 max pool, stride 2
Conv2_x	56*56	$\begin{bmatrix} 1*1, 64 \\ 3*3, 64 \\ 1*1, 256 \end{bmatrix} * 3$
Conv3_x	28*28	$\begin{bmatrix} 1*1, 128 \\ 3*3, \ 128 \\ 1*1, 512 \end{bmatrix} * 4$
Conv4_x	14*14	$\begin{bmatrix} 1*1, 256 \\ 3*3, 256 \\ 1*1, 1024 \end{bmatrix} * 6$
Conv5_x	7*7	$\begin{bmatrix} 1*1, 512 \\ 3*3, 512 \\ 1*1, 2048 \end{bmatrix} * 3$
	1*1	Average pool, 5-d fc

For the detection of weeds species, Faster R-CNN algorithm has been used. The blocks up to Conv4_x were used for Region Proposal Network (RPN) whereas the final block Conv5_x was used as an object detector to predict the class of weeds along with the refinement of the bounding boxes proposed by the RPN. Region Proposal Network consists of a sliding window of kernel size n, over the feature maps generated by the block Conv4_x. At each location of the sliding window, a number of object proposals are computed. These object proposals are computed on the basis of reference boxes, called grid anchors. A number of grid anchors of varying scales and aspect ratios are generated at each position of the sliding window. Though the anchors are generated in feature maps, the coordinates of these anchors corresponds to the image coordinates. The correspondence of the feature maps to that of the image can be calculated as the feature maps are generated through a series of convolutions and pooling.

A grid anchor is labelled as positive if the Intersection over Union (IoU) of the grid anchor to the bounding box of the ground truth is over a certain threshold. Intersection over Union is calculated by the area of intersection of the boxes divided by the area of union. All the grid anchors below the threshold are labelled as negative. These proposals are passed to the classifier and regressor in the RPN where they are classified as objects and background by the classifier and the regressor computes the bounding box regression.

The proposals that have been classified as objects and non-objects along with their regressed bounding box are sent to the detector network, consisting of the final convolution block of the network. Here, the objects are classified into their classes and the coordinates of the bounding box are computed.

3.1 Data Preparation

The images were cropped due to the memory limitations of the computer before being added to the training and test sets. The bounding boxes were created in an XML format using LabelImg tool (LIN 2015). Table 3 shows the number of training and test images used for the study.

Tab. 3: List of Training and Test images

S.N.	Images containing	Number of Training Images	Number of Test images
1	MATCH	78	20
2	POAAN	132	33
3	STEME	174	40
4	VIOAR	126	30
5	AMARE	130	33

The training images contain at least one object and at most 8 objects of an individual class. The bounding box of each object include four coordinates, minimum and maximum x and y coordinate values. In addition, class information is provided.

3.2 Implementation

The weights of the network were initialized by the pre-trained model on COCO dataset and only the weights of the fully connected layer were initialized by Xavier initialization (GLOROT & BENGIO 2010).

The training of the Region Proposal Network (RPN) was done simultaneously along with the training of the object detector, as opposed to the four-step alternating training proposed by REN et al. (2015). A batch size of 1 image was used because the training images vary in their dimensions. Since the training data was sparse, augmentation of images and corresponding bounding boxes was done by randomly flipping them horizontally and vertically, rotating by 90 degrees, altering brightness, contrast, hue and saturation along with random cropping, padding, and scaling.

Grid anchors were generated at scales 0.25, 0.5, 1 and 2 with aspect ratios 1:2, 1:1 and 2:1. The reason for using these is to generate object proposals with varying scales and aspect ratios. This is believed to make the network robust in predicting objects that vary in scale and/or are obscured by other objects. In our case, since the weeds have various stages of development even in the same image and may be obscured by crops, the use of various scales and aspect ratios is important.

For generating the region proposals, a window of size 3x3 was convolved on the results of Conv4_x, followed by the activation step. This output was simultaneously passed to the classification and regression layers by a convolution of 1x1 producing scores for the "objectness" and the four coordinates of a bounding box of each object. The weights of the Conv4_x were initialized by the pre-trained model; however, the weights of the subsequent layers of the RPN were initialized by the truncated normal initialization with a standard deviation σ of 0.01.

The Intersection over Union (IOU) threshold for distinction between objects and non-objects was set to 0.7 which means those proposals whose IOU values were less than 0.7 in the ground truth bounding boxes, were considered as background. As the number of proposals for the background would be particularly high for the images that contain few objects of interest, only a subset of such proposals is considered. In this study, a ratio of 1:1 for the object and background was specified.

The classification of the objects was done by cropping and resizing the convolutional feature maps proposed by RPN and passing it through the Conv5_x layer. The output of this layer was passed through the softmax activation function to predict the class probabilities of the object and the same output was passed to a different regression layer for the final regression of the bounding box. The implementation was done on Tensorflow Object Detection API (HUANG et al. 2016).

3.3 Results and Discussion

Initially, the training was run with up to 200,000 iterations; the accuracy was simultaneously calculated along with the training by evaluating the models created during the training at various iterations using test data. The training loss started from a higher value in the first iteration and then reduced abruptly after few thousand iterations, after that it remained stagnant (see Figure 1). The test accuracy started from a low value and increased abruptly and then remained stagnant after few thousand iterations. Neither the loss nor accuracy improved or degraded even if the training was done for 450,000 iterations (see Figure 1).

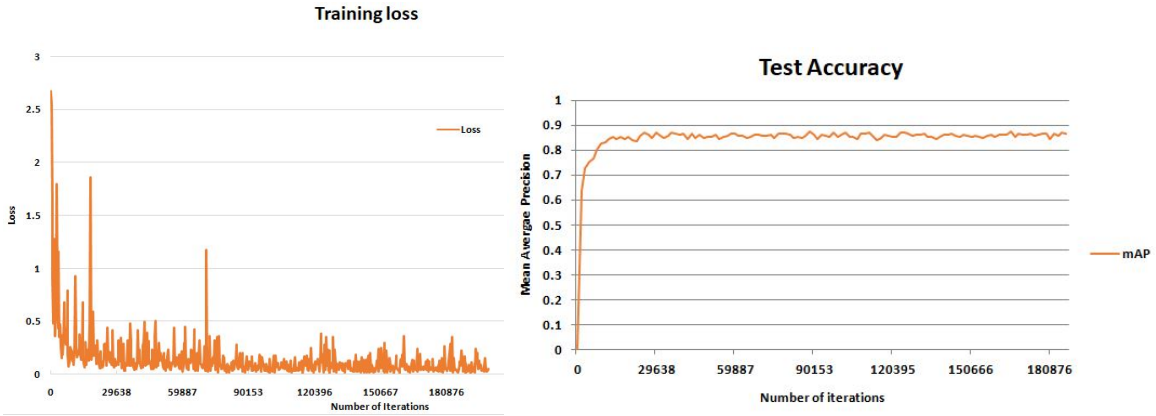


Fig. 1: Training Loss and Test Accuracy

In this setup, the training and test set were mutually exclusive sets of images where the weeds are large. We refer to such images as “large weed images”. It was expected that the network, after sufficient training, would learn the complex features such as shape and texture from the training images and would generalize even on the images where the weeds are significantly small. Hence, another set of such images, which we refer to as “small weed images” were added to the test set to assess the quality of the model. This set of images had 97 images with MATCH and STEME photographed along with wheat. The development stage of the weeds in this image set was similar to the previous set of images, the differences being the size of the weeds, the inclusion of the crops and occurrence of above-specified weeds in a single image.

The evaluation of the detection has been performed according to the PASCAL VOC metrics (EVERINGHAM et al. 2007). The Average Precision (AP) of each class and a mean Average Precision (mAP) for the entire class has been computed. The AP has been calculated by computing precision at various recall values, ranging from zero to one and then taking the weighted mean of such computed precisions, where the weights are the increase in recall values

from the previous steps. The mAP values are the average values of AP over all the classes. The precision is calculated by the number of true positives divided by the sum of true positives and false positives, whereas the recall is calculated by the number of true positives divided by the sum of true positives and false negatives. In our case, true positives refer to the detections that have an Intersection over Union value of at least 0.5. In case of multiple detections of the same class, a single detection is considered as true positive while all the others are considered as false positives. Other false positives are the ones which the model detects although there are no corresponding objects in the ground truth. The false negatives are the ground truth objects which the model did not detect.

Table 4 shows the AP and mAP of each of the weed class which was tested on two different datasets. Dataset 1 refers to the test data set as described in Table 2 and Dataset 2 refers to the test data set including the "small weed images". A third setup (Dataset 3) of the experiment was done in which the "small weed images" were randomly split in training and test data in ratio 80% and 20% respectively, and were added to the respective datasets of Table 2.

Tab. 4: Average Precision and mean Average Precision for three sets of test

	AP(%)					mAP(%)
	AMARE	**STEME**	**MATCH**	**POAAN**	**VIOAR**	**All weeds**
Dataset 1	89.4	74.6	86.3	74.2	96.3	84.2
Dataset 2	87.2	30.2	23.3	40.8	94.0	55.1
Dataset 3	88.5	70.1	78.9	78.9	95.4	82.3

It can be seen that the AP for AMARE and VIOAR does not differ much since the test data of dataset 2 does not contain weeds of small size. However, there has been a massive reduction in AP for STEME, MATCH, and POAAN, and therefore a reduction of the mAP for all weeds. This is due to the wheat in test data images being incorrectly classified as POAAN, as seen in Figure 2 (top row, left). The reason for the misclassification might be the similarities in leaf structure as well as spectral values of wheat and POAAN. Moreover, the size of STEME and MATCH was significantly small. This increased the false positives of POAAN and also the false negatives of STEME and MATCH were increased resulting in low precision and a low recall for the respective classes.

The results show that the accuracy of all the weed species improved and is comparable to the results obtained for the Dataset 1. The justification for the above results comes from the fact that since the "small weed images" are also trained, the Region Proposal Network is able to propose even the smaller objects, shown in Figure 2 (top row, right). Furthermore, inclusion of these images in the training dataset allows the network to assign regions with the wheat to background and hence decrease in false positives for POAAN. The slight decrease in the AP for STEME and MATCH as compared to the first experimental setup might be due to some misclassification among these two classes. The inclusion of "small weed images" in training has negligible effect on the detection of weeds in "large weed images", as shown in Figure 2 (bottom row).

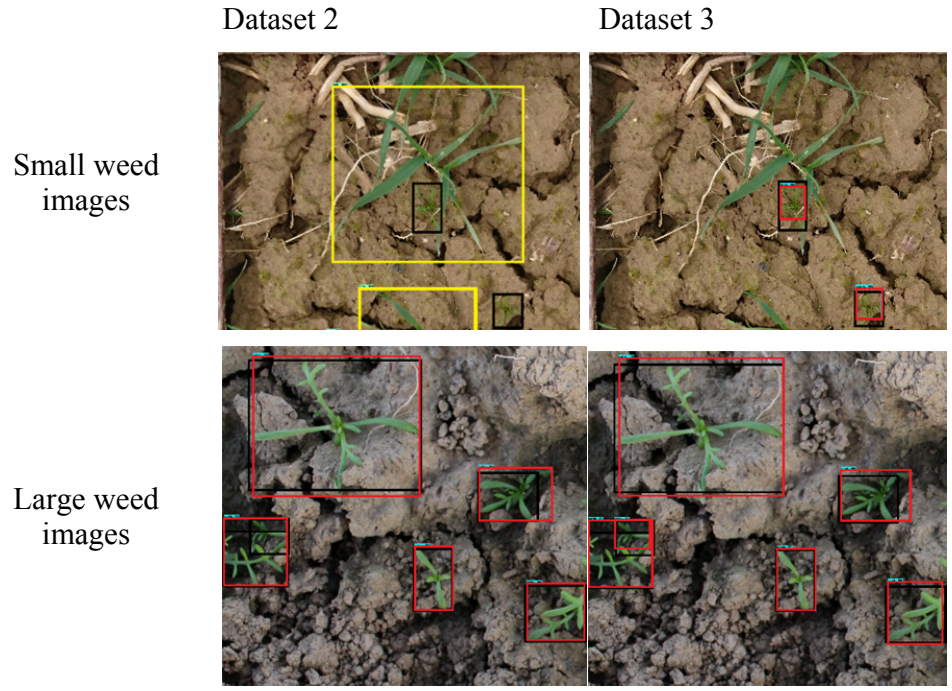


Fig. 2: Detections on test images (network trained on Dataset 2 and 3) with large and small weeds. The ground truth boxes are shown in black. The yellow boxes correspond to misclassification of wheat as POANN and the red boxes correspond to MATCH. The inclusion of "small weed images" for training in Dataset 3 improves the detection but has negligible effect in detection of weeds in "large weed images".

4 Pixel-based approach

Semantic segmentation aims at producing a classification map of the same size as the input image. The map assigns one out of a set of defined classes to each pixel as opposed to categorization networks which assign a single class to an entire image. Several architectures have been created for pixel-wise classification which consists of two main stages, the encoder for classification and the decoder for pixel-wise prediction and output of the segmentation results.

LONG et al. (2014) proposed fully convolutional networks (FCNs) for semantic segmentation. The architecture transforms common classification CNNs such as VGG, AlexNet and GoogLeNet into FCN by rewriting their fully connected layers into convolutional layers which form the encoder stage of the network. The encoder generates low-resolution feature maps which are then passed to the decoder for upsampling to get prediction maps of the same size as the input image. The upsampling is performed using transposed convolution layers also referred to as deconvolution layers through bilinear interpolation. The decoder refines the upsampled outputs by merging them with features from different stages in the encoder stage which are coarse but of high resolution.

Segnet (BADRINARAYANAN et al. 2015), another semantic segmentation architecture, differs from FCN in regards to the implementation of both the encoding and decoding stages. In the encoder stage, all the fully connected layers of a classification network are discarded. The lower

resolution feature maps from the encoder are upsampled in the decoder using unpooling layers as opposed to using deconvolutional layers in the FCN. The resulting encoder-decoder network is efficient in memory usage and computational time. This is due to fewer parameters after discarding the fully connected layers in the encoder and use of max-pooling indices from the encoder for non-linear upsampling in the decoder. The low memory usage, however, results in loss of accuracy compared to FCN which preserves the feature maps in the encoder stage by re-writing the fully-connected layers instead of discarding them. For this reason, FCN was used in this approach.

4.1 Data Preparation

For training a FCN, image pairs as data input and a corresponding ground truth mask as the label input is required. Tab. 5 shows the number of training image samples created from the field images and the plant seedlings dataset (GISELSSON et al. 2017)

Tab. 5 : The training Dataset

EPPO Code	Name	No. of Images	No. of Images + Augmentations
MATCH	Wild chamomile	179	1611
POAAN	Annual meadow grass	212	1908
STEME	Common chickweed	200	1800
VIOAR	Field pansy	168	1512
	Wheat	201	1809
TOTAL		960	8640

The image samples were cropped to contain exactly one weed or wheat and the background unlike in the object-based approach in Section 3.1 where a training image contained one or more plants. Therefore, the number of images for each species in this approach is higher than the one used in the former approach. The reason for cropping in the indicated way was to allow segmentation through maximum likelihood classification using ArcGIS Desktop. Symbolic values corresponding to the six classes, five classes in the training dataset and the background, were assigned to the pixels of the segmented images. These symbolic images were used in the sequel as ground truth labels. In order to expand the training dataset effectively, data augmentation was performed where both geometric and pixel value changes were applied. Increasing the dataset has been used to reduce overfitting and speed up convergence while training models. The geometric augmentation techniques included rotating by 90º, 180º and 270º, horizontal and vertical flips of both the images and the labels. In the second set of augmentations, the pixel values were altered by performing colour enhancement, colour equalization and applying a Gaussian blur. Unlike in the object-based approach where the augmentations were done on the fly, these augmentations were performed prior to the training process. The image dataset was split with 80% as the training dataset and 20% as the test dataset for each class and stored in light memory-mapped databases (LMDBs).

4.2 Implementation

The FCN architecture was implemented by leveraging models trained on bigger datasets through transfer learning. The deep learning framework Caffe (JIA et al. 2014), developed by the Berkeley Vision and Learning Centre (BVLC), was used for the implementation while the model definitions and pre-trained weights were obtained from Caffe's Model Zoo.

In the first set up, a VGG-16 model by SIMONYAN AND ZISSERMAN (2014) used for the ImageNet Challenge 2014 was transformed into a fully convolutional network. The VGG-16 network has three fully connected layers at the end, FC6, FC7 and FC8. The last layer FC8, used as the classifier for ImageNet's 1000 classes, was discarded while FC6 and FC7 were re-written into convolutional layers and the weights transferred using the net surgery script provided in Caffe. By reshaping and retaining FC6 and FC7, the feature extraction weights from the VGG-16 are preserved. The weights were used to initialize the encoding stage of the "at-once" FCN-8s architecture used for training with PASCAL VOC 2011 dataset (EVERINGHAM et al. 2011). Gaussian and bilinear weight fillers were used to initialize the weights of the convolutional and deconvolutional layers respectively in the decoder. A combination of a low learning rate of 10^{-10}, a high momentum of 0.99 and a batch size of one was used for the training. This setup is referred to as VGG16-FCN in the next sections.

In the second experimental setup, the weights of a FCN-8s model trained on 20 classes of the PASCAL VOC dataset were fine-tuned using the weeds dataset. This was done by a full transfer of weights and adding a convolution layer as a classifier with the six classes. The new classifier was initialized with a Gaussian weight filler. A low learning rate of 10^{-14} has been used together with a high momentum of 0.99 to fine-tune the entire network. The low learning rate was necessary to prevent fast distortions to the fully transferred weights.

4.3 Results & Discussion

In the training phase, choosing and optimizing training parameters required monitoring by plotting the training loss, test loss and accuracy which indicate the performance in real time. The goal of this is to decrease the training and test losses while increasing the model's accuracy. The learning rate remains the most important parameter and estimating an optimal value involves testing a range of values and different learning rate policies. Very low learning rate means the training takes long to converge or gets stuck at a local minimum while a very high learning rate leads to the loss function rising very fast. In both setups, a combination of a fixed learning policy rate, a low learning rate and a high momentum gave better results.

Caffe offers the flexibility to specify which layers to either fine-tune or freeze the weights during training. Fine-tuning the whole network gave better results which can be attributed to the difference in nature of training classes of our dataset from the ImageNet (RUSSAKOVSKY et al. 2014) and PASCAL VOC (EVERINGHAM et al. 2011) datasets used to train the VGG-16 and FCN-8s models. Attempt to perform training without initializing new layers with weight fillers resulted in high loss values. Gaussian and Xavier weight fillers were both tested to initialize new convolution layers both yielding similar performance.

The performance of the models from the two experiments was evaluated on the test dataset using mean Intersection over Union (mIU) metric. The metric also known as Jaccard similarity coefficient is the ratio of correctly classified pixels to the total number of ground truth and

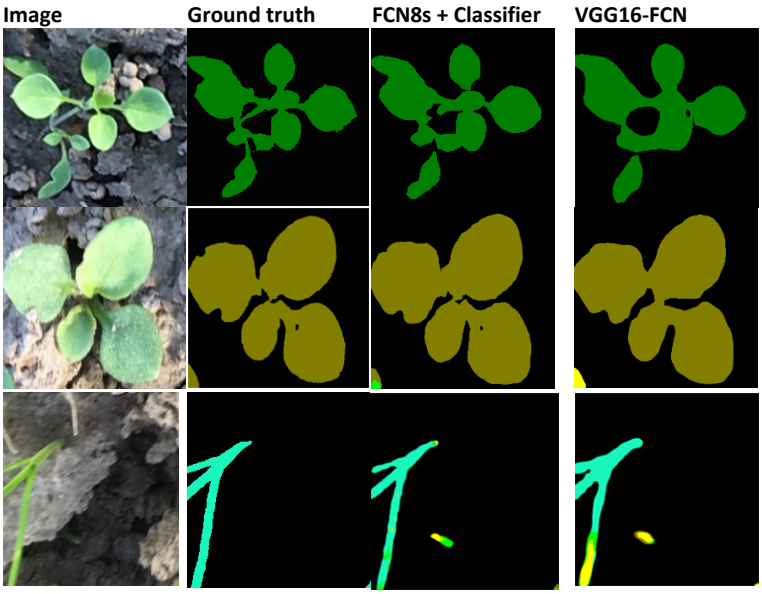
predicted pixels. It provides an accurate measurement that penalizes false positives making it suitable for evaluating semantic segmentation models. The pixel accuracy can be a misleading measure of model performance if one class has a higher number of pixels in the images compared to other classes. In the training and test datasets, most of the pixels belong to the soil which is represented in the class background. The frequency weighted IU, considers the IU of each class weighted by the number of pixels in that class. This gives an overall accuracy with larger classes receiving larger weights hence the high results in Tab. 6, due to the influence of the background class which does not give an accurate performance of the models. The models predict most of these background pixels correctly but misclassify some of the weed and wheat pixels. In addition to the metrics, visualization of the segmented images was compared to the ground truth labels.

- Mean accuracy: $(\mathbf{1}/\boldsymbol{n_{cl}})\sum_i \boldsymbol{n_{ii}} / \boldsymbol{t_i}$
- Mean Intersection over Union (IU): $(\mathbf{1}/\boldsymbol{n_{cl}})\sum_i \boldsymbol{n_{ii}} / (\boldsymbol{t_i} + \sum_j \boldsymbol{n_{ji}} - \boldsymbol{n_{ii}})$
- Frequency weighted IU: $(\sum_k \boldsymbol{t_k})^{-1}\sum_i \boldsymbol{t_i}\,\boldsymbol{n_{ii}} / (\boldsymbol{t_i} + \sum_j \boldsymbol{n_{ji}} - \boldsymbol{n_{ii}})$

Here $\boldsymbol{n_{cl}}$ is the number of classes, $\boldsymbol{t_i}$ total number of pixels of class $\boldsymbol{i}$ in the ground truth label, $\boldsymbol{n_{ii}}$ is the number of correctly predicted pixels of class $\boldsymbol{i}$, and $\sum_j \boldsymbol{n_{ji}}$ is the total number of pixels predicted to belong to class $\boldsymbol{i}$ (LONG et al. 2014).

Tab. 6 : A comparison of evaluation metrics after 30,000 training iterations of each experimental setup calculated on the test dataset with 1,725 Images

Models	Mean accuracy	Mean IU	Frequency weighted IU
VGG16-FCN	77.60	24.75	91.7
FCN8s + Classifier	77.43	25.32	92.84



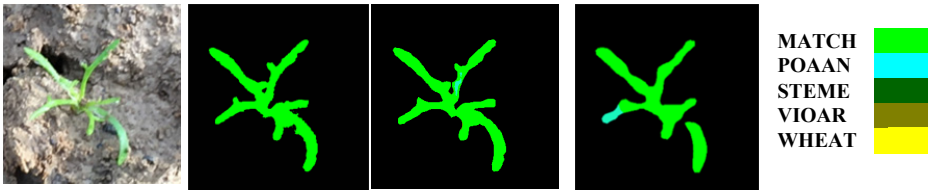


Fig. 3: Visual comparison of segmentation results of images from the test dataset taken in the same conditions as the training dataset

From the metrics in Tab. 6 and inference results in Fig. 3, the *FCN8s + Classifier* setup yields slightly better results. This method employs full weights transfer with only one layer being randomly initialised compared to the first method where the entire decoder weights are randomly initialized. Weeds sharing similar leaf structures such as elongated leaves in classes POAAN, Wheat and MATCH are susceptible to misclassifications. This can be seen in Fig. 3 where some POAAN pixels get misclassified as wheat or MATCH and the same occurs in MATCH segmentation. The models were further tested on images from the plant seedlings dataset (GISELSSON et al. 2017) as a control check for overfitting. The results in Fig. 4 show the ability of the models to predict a majority of the pixels correctly with the *FCN8s + Classifier* setup yielding better spatial detail and less misclassified pixels.

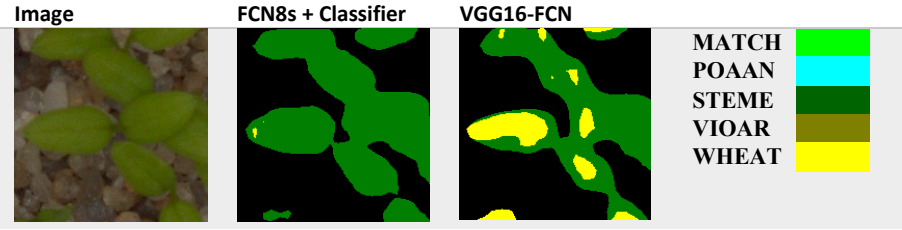


Fig. 4: Segmentation results for a STEME image from the plant seedlings dataset

5 Conclusion

The experiments have demonstrated success in transferring pre-trained CNNs to address weed classification with limited training images. Further improvements could be achieved by expanding image datasets. A large collection of weed and crop images varying in growth stages, soil textures, resolutions and conditions would help in training robust CNNs for use in precision agriculture.

The resolution of training images poses some restrictions both due to the memory requirements for the computational process as well as in inferring the generated model. The models trained in images where the weeds are large, have difficulties in detecting the weeds when the size of the weeds is significantly small. However, the inclusion of some images with small weeds in training seems to improve the performance of such models. Further experimentation is required to be able to detect objects that vary significantly in scale as compared to the trained objects so as to create more robust models.

There are still challenges to detect objects when they are obscured by other objects. Various parameters such as initialization of the weights, regularization of the network affect the learning process and have to be further investigated. Other semantic segmentation networks have been proposed either as extensions to the FCN models or entirely different architectures focusing on reducing computational resources and time, increasing accuracy or a combination of the three. These could be explored and compared.

6 Acknowledgements

The project was supported by funds of the Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the innovation support programme.

7 References

ÅSTRAND, B. & BAERVELDT, A.J., 2002: An Agricultural Mobile Robot with Vision Based Perception for Mechanical Weed Control, Autonomous Robots, **13**, 21-35.

BADRINARAYANAN, V., KENDALL, A. & CIPOLLA, R. 2015: SegNet. A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv:1511.00561 [cs.CV].

EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C.K.I., WINN, J. & ZISSERMAN, A., 2007: The Pascal Visual Object Classes Challenge 2007 (VOC 2007) results (2007), available at: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C.K.I., WINN, J. & ZISSERMAN, A., 2011: The PASCAL Visual Object Classes Challenge 2011 (VOC 2011) Results, available at: http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

GISELSSON, T.M., DYRMANN, M., JØRGENSEN, R.N., JENSEN, P.K. & MIDTIBY, H.S., 2017: A Public Image Database for Benchmark of Plant Seedling Classification Algorithms. arXiv:1711.05458 [cs.CV].

GLOROT, X. & BENGIO, Y., 2010: Understanding the Difficulty of Training Deep Feedforward Neural Networks, in Teh, Y.W. and Titterington, M. (Eds.). Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR, Chia Laguna Resort, Sardinia, Italy, 249-256.

GOLZARIAN, M.R. & FRICK, R.A., 2011: Classification of Images of Wheat, Ryegrass and Brome Grass Species at Early Growth Stages Using Principal Component Analysis, Plant Methods, **7**(1), 28.

HE, K., ZHANG, X., REN, S. & SUN, J., 2016: Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

HUANG, J., RATHOD, V., SUN, C., ZHU, M., KORATTIKARA, A., FATHI, A., FISCHER, I., WOJNA, Z., SONG, Y., GUADARRAMA, S. & MURPHY, K., 2016: Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. arXiv:1611.10012 [cs.CV].

JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R., GUADARRAMA, S. & DARRELL, T., 2014: Caffe: Convolutional Architecture for Fast Feature Embedding, ACM, available at: http://dl.acm.org/ft_gateway.cfm?id=2654889&type=pdf.

LIN T., 2015: LabelImg. Github Repository, https://github.com/tzutalin/labelImg, Accessed on 01/27/2018.

LIN, T.-Y., MAIRE, M., BELONGIE, S., HAYS, J., PERONA, P., RAMANAN, D., DOLLÁR, P. & ZITNICK, C.L., 2015: Microsoft COCO. Common Objects in Context. arXiv: 1405.0312 [cs.CV].

LONG, J., SHELHAMER, E. & DARRELL, T., 2014: Fully Convolutional Networks for Semantic Segmentation. arXiv:1411.4038 [cs.CV].

REN, S., HE, K., GIRSHICK, R. & SUN, J., 2015: Faster R-CNN. Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing, 91-99.

RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M.S., BERG, A.C. & LI, F.-F., 2014: ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575 [cs.CV].

SIMONYAN, K. AND ZISSERMAN, A., 2014: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs.CV].

SØGAARD, H.T., 2005: Weed Classification by Active Shape Models. Biosystems Engineering, **91** (3), 271-281.