Hemp Segmentation and Classification, a Deep Learning approach

Schori Damian
Vertiefungsarbeit 2, MSE
ILT, Hochschule für Technik Rapperswil

damian.schori@hsr.ch

Abstract **– The yield of crops such as hemp (Cannabis) depends on various factors, such as environmental influences, location site and lighting conditions. Therefore, the task of quantifying the volume and the number of plants per unit area as well as to classify different species in the same field is critical for plant breeders to get an exact overview of the current situation in the field. For large scale sites it is almost infeasible to manually count and estimate the volume of such fields.**

To automate this process, a machine learning approach, based on deep convolutional neural networks (CNN), is proposed to generate pixel level segmentation maps for area and volume calculations and to estimate the number of plants on non-overlapping cases from UAV- images.

It could have been shown that a reasonable number of training samples already yield good results. With post-processing steps such as majority voting which is taking shots on multiple dates into account the method delivers good results on images where it is difficult even for human individuals to label the plants correctly.

The overall segmentation rate is above 86% measured by the dice coefficient and an accuracy of about 84.5 % and higher regarding the plant counting is reported. The developed solution is therefore a robust method to segment and detect Hemp plants under real-world conditions.

Keywords **- Deep Learning, Image Segmentation, Computer Vision, Remote Sensing, Hemp Plants, Cannabis**

1 GLOSSARY

Word	Description
Subseed	The seed, which is growing between the hemp plants.
Species 1001	Hemp plant of species 1001
Species 1005	Hemp plant of species 1005
Field C	Field used for training and validation
Field A	Field used for testing (model has not seen this during training)
QGIS	Quantum GIS, is a free and open-source cross-platform desktop geographic information system
CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
IOU	Intersection over union
RGB	Red- Green- Blue images
NIR	Near infrared images
VIS	Visible infrared images
DSM	Digital surface model
UAV	Unmanned aerial vehicle

2 INTRODUCTION

Hemp (cannabis) is a plant genus of flowering plants in the family of the Cannabaceae and is one of the oldest cultivated and ornamental plants on earth. Hemp is a mostly annual plant and reaches, depending on environmental conditions, very different growth heights that can reach up to 5 meters depending on the nutrient supply.

Since the sale of cannabis with a low THC content was approved in Switzerland at the end of 2016, the number of hemp producers has risen sharply [1] . In these plants, however, the cannabidiol (CBD) remains, which is used to treat cramps, inflammation and nausea [2] and is also consumed as a stimulant.

Due to the strong increase of hemp production in Switzerland, the competition has also increased, which in turn has an impact on prices. This requires the ability to cultivate hemp plants efficiently in order to be able to produce cost-covering even in a highly competitive market. The large-scale cultivation requires a constant and precise monitoring of the fields in order to supervise the growth of the plants in the best possible way and to intervene at the right moment with suitable tools if necessary. It is also important to be able to optimally plan the logistics during harvesting. This ensures that an optimum number of harvesters and machinery can be requested.

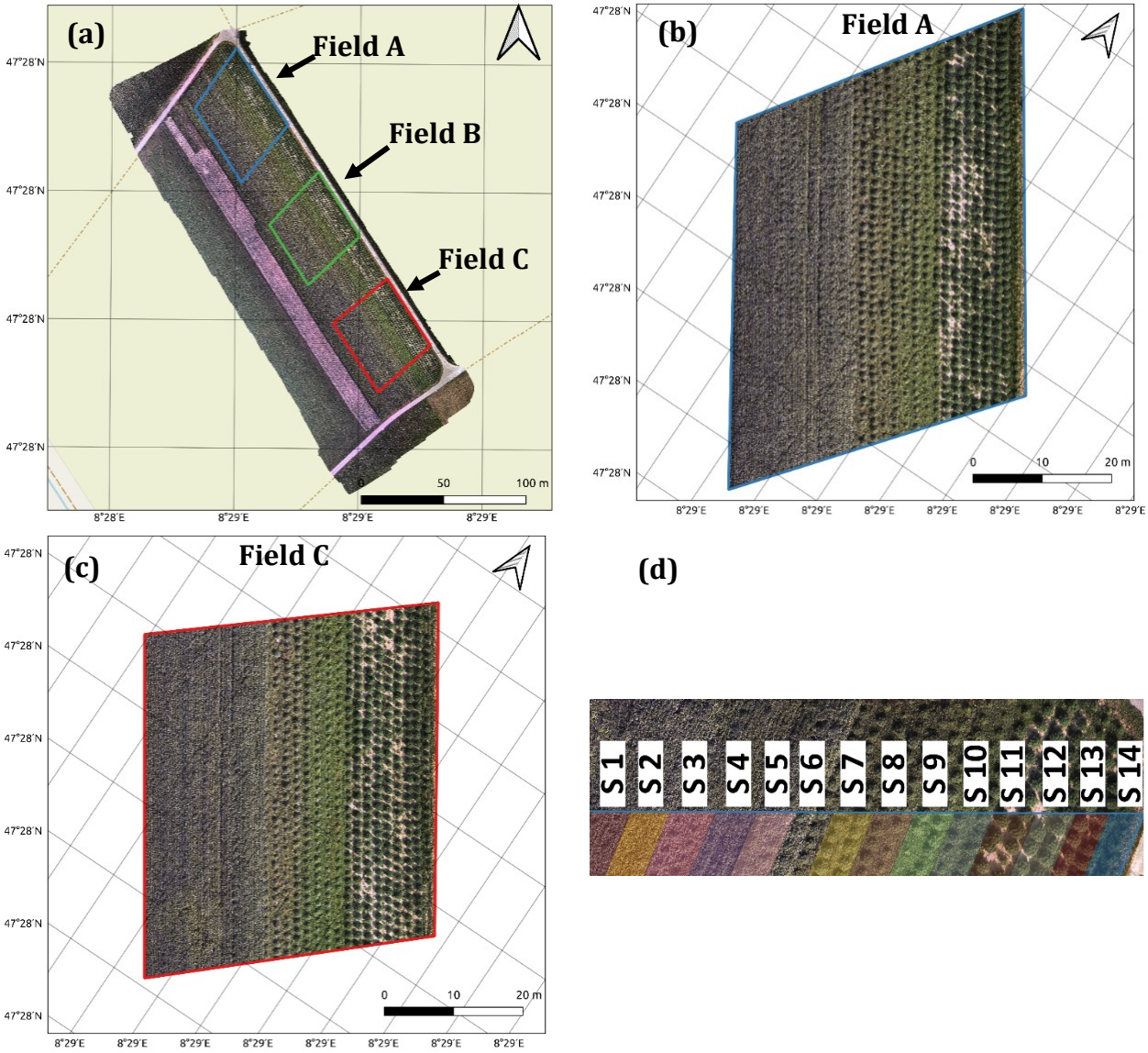

Fig. 1: Geographical setting of the test area. The area is located near the village of Niederhasli, in the area of Zürich (**a**). The area is subdivided into 3 fields (A, B, C) whereas only Field C (**c**) for training and Field A (**b**) for testing where used. In (**d**), the subdivision of the different subseeds 1-14 is visible.

2.1 Related works

Machine learning and image processing have proved their utility in diverse fields. Especially in the field of plant phenotyping [3, 4, 5, 6, 7] these tools have laid a strong foundation in detecting multiple crop diseases [8] as well as making sense of disease severity without the need for any additional human supervision [8], crop/weed discrimination [9, 10, 11, 12], canopy/individual extraction [13, 14], fruit counting/flowering [15, 16, 17], and head/ear/panicle counting [18, 19, 20, 21].

More recent applications have emerged which attempt to identify and quantify individual plants. In [22] banana plants are detected and counted with the help of Convolutional Neural Networks and very good results are achieved using only RGB images. In [21] a method for detecting and counting sorghum heads is described where a RetinaNet [23], which is also based on Convolutional Neural Networks, is used. Both [22, 21] are based on a recognition and counting task, but do not allow segmentation and thus area and volume calculation.

Another application is the detection of illegally planted hemp for law enforcement purposes. For example, plants with a higher THC content can easily be hidden in existing legal hemp fields. In order to identify illegal hemp plantations, attempts were made to distinguish hemp from non-hemp by means of

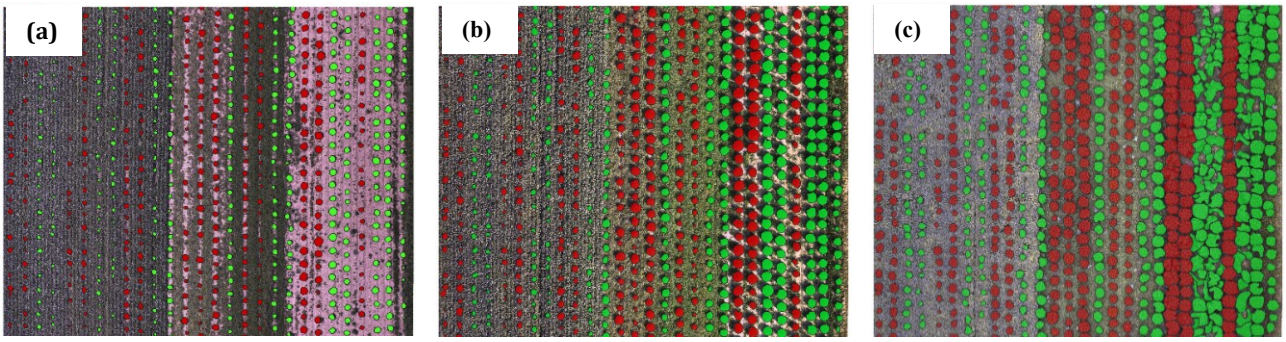

Fig. 2: Three samples of Field C on the dates: 03.07.2019 (**a**), 19.07.2019 (**b**) and 22.08.2019 (**c**) with labeled Plants corresponding red to Species 1001 and green to Species 1005.

hyperspectral images in the visible near-infrared region (VNIR) [24].

2.2 Goal

The hypothesis of this work is that machine learning and image processing along with unmanned aerial vehicles (UAV) based photogrammetry is a reliable alternative to the time consuming plant survey in the field. With the assistance of UAV it is nowadays possible to take field recordings with simple equipment. The goal is to develop a method to detect and classify two different hemp species and to quantify them in terms of their volumes, quantities and species. For this experiment, images were taken with red-green-blue (RGB) cameras as well as near-infrared (NIR) and visible-infrared (VIS) cameras on a test field in Niederhasli (ZH) with the help of a drone. The test field is divided into three subfields (Fig. 1), two of which are used for training purposes and one for testing. These subfields are further subdivided into 14 sections with different seeds (Fig. 1, d). The RGB-NIR-VIS images were acquired at 11 dates from May until October 2019. For this work, however, only RGB (read in chapter 8.1 why)recordings of three dates were used (03.07, 19.07 and 22.08).

The research contribution will be to develop a method by which hemp plants can be identified, classified and segmented in a robust manner. Furthermore, post-processing steps will show how the accuracy can be improved by using a majority voting system over multiple data and centre crop images.

3 METHODOLOGY

3.1 Study Area and Image Data

From the three fields two were used, one for training and validation (Field C) and one for testing (Field A) with a corresponding area of 2071m^2 and 2304m^2 respectively. The fields are each divided into 14 subseed- zones as shown in Fig. 1:

- Subseed 1 – 2: Soybean
- Subseed 3 - 4: Soybean + Clover
- Subseed 5 - 6: Soybean + Clover + Meadow fescue
- Subseed 7 – 8: Clover + Meadow fescue
- Subseed 9 – 10: Clover
- Subseed 11 – 14: No Subseed

The images were then stitched and georeferenced using the open source geoinformation system QGIS.

3.2 Labeled Data

To get ground truth data for training, validation and testing approximately 6400 hemp plants were hand-labeled with their corresponding class and outer contour in QGIS [25]. This consisted of a polygon object for each individual plant containing their class as an attribute. To create the training, validation and testing sets, a grid was created with a grid size of 512 x 512 pixels for training and 384 x 384 pixels for testing. The larger size of the training images allowed to sample randomly croped images in the size of 384 x 384 pixels instead of padding them for translation augmentation. For further processing and model training the georeferenced maps were then sliced into 384 x 384 pixel patches (approximates to 3 x 3 meters) from both field A and C. The dimensions of the labeled data were chosen based on the two following considerations: (1) to

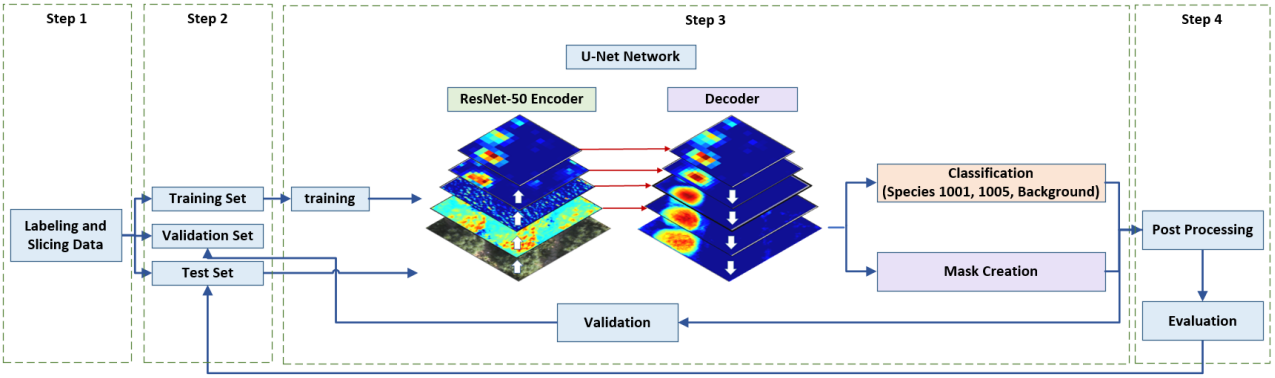

Fig. 3: General Workflow

maximize the number of slices per subset, as a bigger resolution would result in less samples and higher feature space. And (2) as pre-trained networks are used for transfer learning which are pre-trained mostly on resolutions about 384 x 384 pixels or lower. The training and validation data sampled from Field C were then randomly split into 80% for training and 20% for validation. For further processing the labeled polygon masks are rastered into pixel-based images with 3 channels corresponding to the classes (Species 1001, Species 1005, background) [height, width, class].

3.3 Problem Description

The segmentation and detection of individual plants in their natural environment is a challenging task due to plants showing significantly varying poses, sizes and complex shapes in natural environmental conditions.

Depending on the 14 different subseeds which grow between the hemp plants, there are strong differences in how fast the hemp plants grow. This has the consequence that on a section of the field where no subseed had been planted, the hemp plant already stands out strongly from the ground (Fig. 5). On the other hand, on sections of the field where a strongly growing subseed has been sown and the hemp plants do not yet stand out from the subseed, it is difficult to recognize them (Fig. 4). Additional factors such as changing light and field properties influence the detection.

Fig. 4: Detail of a training image containing hardly visible hemp plants

Fig. 5: Detail of a training image containing clearly visible hemp plants

Moreover, the different species show only few distinguishing features. Species 1001, for example, is usually somewhat darker and the leaves are softer than those of species 1005 (Fig. 6).

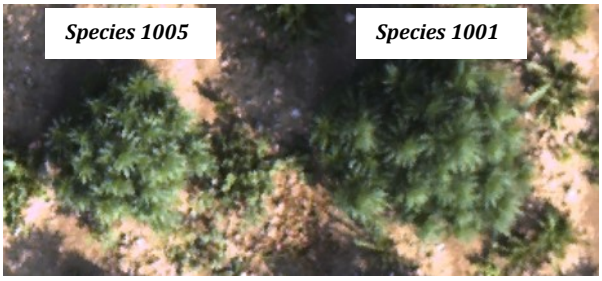

Fig. 6: Different Species

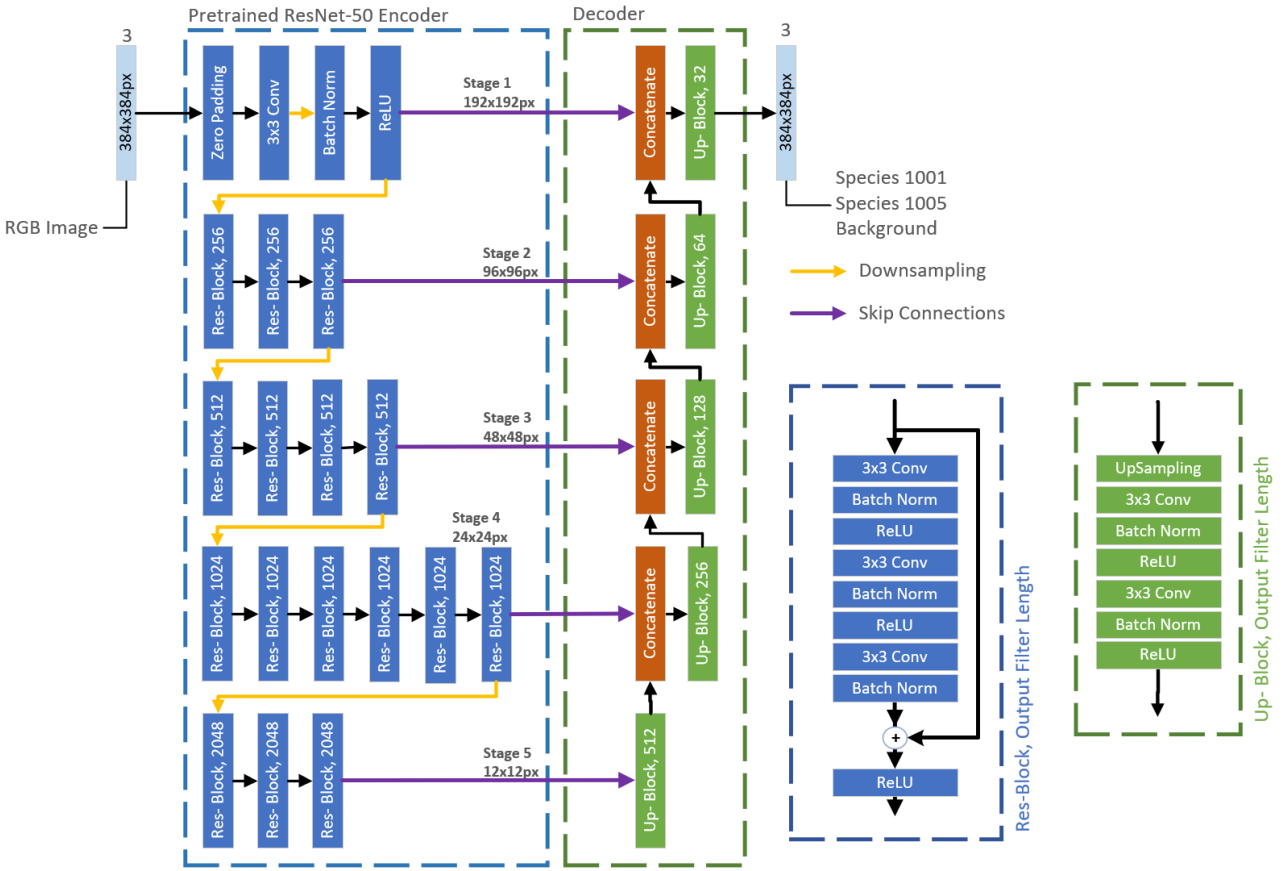

Fig. 7: Modified U- Net: The proposed deep learning architecture

3.4 General Workflow

The General Workflow of the Hemp detection and classification rests on four key steps (Fig. 3): (Step 1) Label and slice the data for training and evaluation. (Step 2) Division of the RGB imagery into patches for training, validation and testing as described in. (Step 3) Training the U-Net Algorithm with the training set and validation on the validation set. (Step 4) Applying post processing steps and evaluating on the test set.

4 PROCEDURE

4.1 Deep Learning Framework

The proposed hemp segmentation scheme is mainly making use of an adapted version of U-Net (Fig. 7), a Fully Convolutional Neural Network (FCNs) proposed in [26]. The U-Net like Network was chosen due its simplicity and effectiveness on segmentation task.

FCNs were introduced [27, 28, 26] in the literature as a natural extension of CNNs to tackle per pixel prediction problems such as semantic image segmentation. FCNs add up-sampling layers to standard CNNs to recover the spatial resolution of the input at the output layer. Consequently, FCNs can process images of arbitrary size. In order to compensate the resolution loss induced by pooling layers, FCNs introduce skip connections (see Fig. 7) between their down-sampling (encoder) and up-sampling (decoder) paths. Skip connections help the up-sampling path recover fine-grained information from the down-sampling layers.

The proposed network features an encoder path with five resolution levels (stages), which reduce the spatial dimensions of the input image from 384x384 to 12x12 pixels at the lowest resolution and encode the relevant context and information into 2048-dimensional feature vectors. For the Encoder a ResNet50 [29] Architecture was chosen. The ResNet50 is pre-trained on more than a million images from the ImageNet database [30] and has therefore learned a rich feature representation for a wide range of images. The decoding path is used to enable precise localization of the plants using up-sampling and convolution layers.

4.1.1 Convolutional Networks

In the case of 2D image data, Convolutional Neural Networks (CNNs) have proven to be state of the art when it comes to the point of image analysis. CNNs convolve two dimensional input images with kernel matrices which are learned during the training process. This allows the network to learn local patterns, which start mostly with low level features like edges and end up in this case with the location of whole hemp-plants (see chapter 5.4). This behaviour gives the network two important properties:

- The learned patterns are (mostly) translation invariant. Because the kernel is shifted over the whole image, a learned pattern is recognized everywhere in the image. This is very helpful in this case because plants appearing everywhere in the images.
- CNN's are learning spatial hierarchies of a pattern. Usually the first layers are learning simple low level patterns such as edges, corners or angles. Deeper layers start to learn more complex features such as leaves in this case. With this, CNNs can very efficiently learn increasingly complex and abstract visual concepts and therefore find patterns where conventional methods often fail.

4.1.2 Environmental influences

Another advantage of deep neural networks is the ability to adapt to almost any data. So, if we want the network to make predictions on data from a specific distribution we "just" have to train the network on this type of data. Therefore, it is possible to address environmental influences mentioned in Chapter 3.3 by sampling and training on exactly those situations. The network is then able to make predictions on new data taken under similar conditions. But this is also the point. As soon as data from a different distribution appears at test time, the model will no longer provide the same results and thus may fail in recognition.

Since it is usually difficult to acquire and label enough data for all required situations, there is the possibility of data augmentation. This can help to fill data gaps and to enlarge the data set in general.

4.2 Data Augmentation

The network has a large number of trainable parameters (about 40 million) compared to the amount of available training images (432 images). To teach the network the desired invariance and robustness properties real time data augmentation [31] during training time was applied.

In the case of field-images mostly shift and rotation invariance as well as robustness to light, size and bluring variations is needed. The number of augmentation operations applied is chosen randomly from zero to four operations per image. To achieve those properties the following augmentations were applied:

- Random Crop
- Random Flip
- Random Brightness
- Gaussian Noise

4.3 Implementation

The U- Net Architecture (Fig. 7) was implemented using the open-source package Tensorflow 2.0 [32]. The codes are available on Github (https://github.com/dschori/Hemp-Segmentation). To perform the experiments a single computer with an Intel Xenon CPU, 32 GB RAM and a GeForce GTX1070 graphics card was used. In the training process the graphics processing unit (GPU) was used to train the U-Net with a batch size of 8 images per step, 43 steps per epoch and an initial learning rate of 0.0002 with a learning rate decay of 0.3 every 10th epoch for 100 epochs.

4.4 Model Optimization

The U-Net has been optimized with a combination of categorical- crossentropy and dice coefficient:

$$H_{BCE} = -\sum_{i=0}^{M} (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i))$$

$$H_{Dice} = -\frac{2 \cdot \sum_{i=0}^{M} (\hat{y}_i \cdot y_i)}{\sum_{i=0}^{M} (\hat{y}_i + y_i)}$$

where:

$$H = H_{BCE} + H_{Dice}$$

- With $i = \{0 \,..\, M\}$ denote the indices of the flatted vectors of y and $\hat{y}$.containing all (three) classes
- With y_i denotes the ground truth mask
- With $\hat{y}_i$ denotes the predicted mask

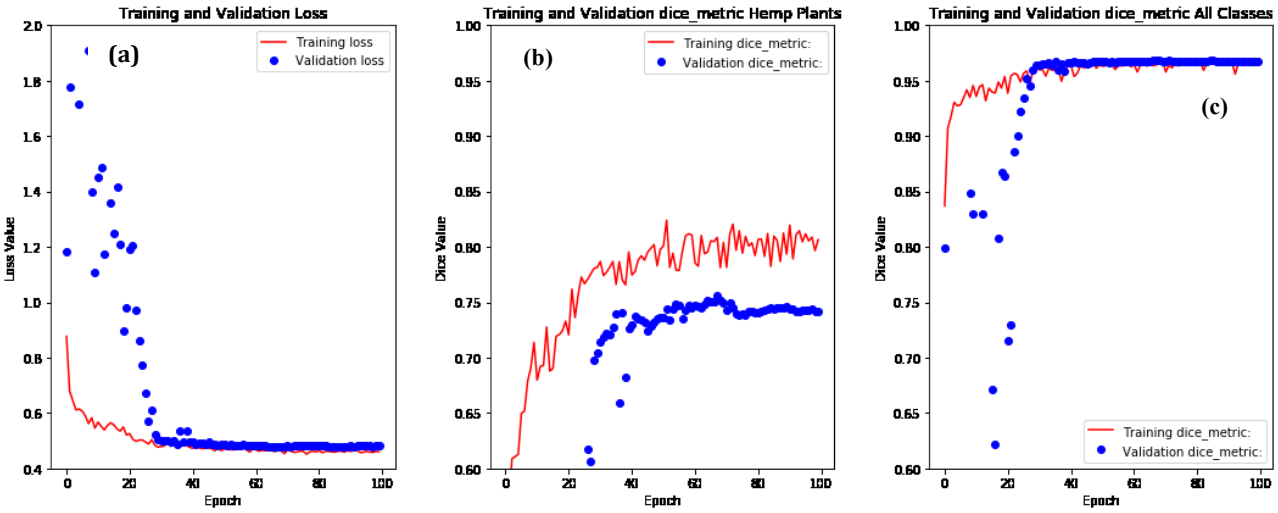

Fig. 8: Value of the loss function (**a**) and the two metrics which monitor accuracy regarding only hemp plants (**b**) and hemp plants + background (**c**) during training.

In addition, individual weights were assigned to the respective classes in order to control the importance of the different classes. A higher weighting was assigned to the background in order to increase the weighting and penalize the boundaries between individual instances. In addition, a metric was set up to monitor the accuracy during the training process. It measures the Dice Coefficient between the two hemp classes 1001 and 1005 but not from the background as this is not decisive for the final result:

4.5 Metrics

To check if the model fits to the data during training, two metrics were defined which monitor the training process and provide information about the current accuracy of the model. These metrics monitor the accuracy after each training and validation step (Fig. 8). They monitor (1) the accuracy of both hemp species without background and (2) the accuracy of all classes (both species + background). This provides insight into how the model behaves during training and whether an over- or under-fitting is occurring. Furthermore, callbacks can be used to react to the training process and, for example, the model can be stored temporarily after each improvement towards the validation set.

$$(1)\ M_{Hemp} = -\frac{2 \cdot \sum_{i=0}^{M}(\hat{y}_i \cdot y_i)}{\sum_{i=0}^{M}(\hat{y}_i + y_i)}$$

- With y_i denotes the ground truth mask
- With $\hat{y}_i$ denotes the predicted mask
- With $i = \{0 \,..\, M\}$ denote the indices of the flatted vectors of y and $\hat{y}$ containing only the hemp classes

$$(2)\ M_{All} = -\frac{2 \cdot \sum_{i=0}^{M}(\hat{y}_i \cdot y_i)}{\sum_{i=0}^{M}(\hat{y}_i + y_i)}$$

- With y_i denotes the ground truth mask
- With $\hat{y}_i$ denotes the predicted mask
- With $i = \{0 \,..\, M\}$ denote the indices of the flatted vectors of y and $\hat{y}$ containing all (three) classes

4.6 Training

The model was trained for 100 epochs and validated after each epoch on the validation set of Field C. After each improvement towards the validation set, the model was saved and after training finished the best model has been restored. During the training process, the metrics (Described in chapter 4.5) and the loss value were recorded (Fig. 8). The figure shows that the validation curve for the metric which measures the accuracy of all classes (c) adapts well to the training curve, which means that there is little or no over- and under-fitting. Based on this it can also be concluded that the model has a suitable number of parameters to fit the training set.

5 Results

5.1 Prediction Threshold

After the training process, predictions can be made on the test set. The model returns a mask for each input image with a probability reaching from 0

to 1 noted in each pixel for every class. However, a mask with Boolean values is required for further processing. To obtain this, each pixel can be threshold after a defined value and is then either counting to a specific class or not. The optimal threshold value can be determined empirically using the validation set.

Therefore, the validation set is used to make predictions for different thresholds, for which the respective accuracy can be determined based on the ground truth. The best threshold is then used to calculate the predictions on the test set (Fig. 9).

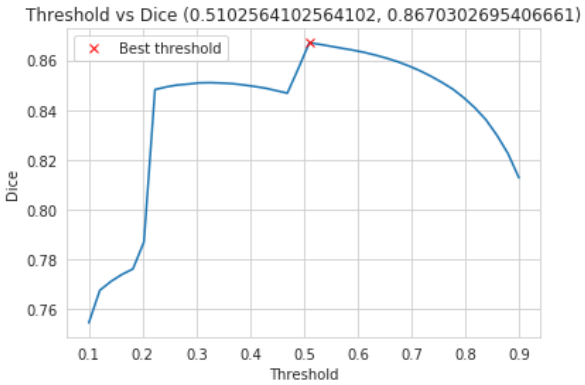

Fig. 9: Threshold optimization

Based on Fig. 9, the optimal Threshold-Value has been found at 0.51 resulting in a Dice Coefficient on the validation set of 0.87.

5.2 Mapping

Until now the sliced images were used for processing. To be able to check the accuracy of the whole field, the individual images are stitched together to form a complete image which can then be georeferenced.

This was achieved by creating a function which allowed to fade over the field in a "sliding window" manner. Thereby predictions are made on the individual windows and are then "replaced" with the input window. This allows the proposed post-processing steps to be performed on the entire field. After the process is finished, the "prediction- maps" are then saved as georeferenced .tif files.

5.3 Post Processing and Accuracy

To improve accuracy of the base model, two post-processing steps are applied:

- Based on the recordings of three different dates, a majority voting system is introduced.
- Further, it can be shown that most mistakes are made on border regions of images which can be fixed by central cropping the images.

To analyse the accuracy of the model and the proposed post-processing steps, the following two points are analysed:

- Visual inspection of six selected sections on the test set (field A) Fig. 10.
- Checking the dice coefficient on all three dates

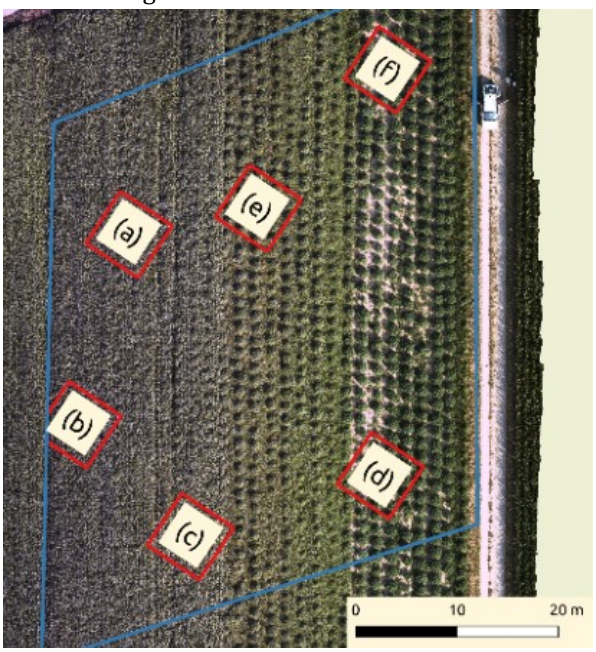

Fig. 10: Selected sections (a...f) on Field A for visual inspection

5.3.1 No Postprocessing

To be able to compare the following post-processing steps, a baseline- map was made for all dates using only the model.

If we look at the visual results in Fig. 13, it is obvious that many plants on 03.07 are wrongly classified (b), (c), (e), (f). Furthermore, on the date of the 19.07 some artefacts which are caused by stitching the predictions are visible.

5.3.2 Majority Voting

Looking at graphs (b) and (c) in Fig. 8, one can see that most mistakes are not made in the general segmentation of hemp plants but in the class-specific segmentation. Therefore, it can be said, that the highest error rate occurs in the classification of the respective species. With the help of the three different dates (03.07, 19.07 and 22.08) it is now possible to compare each individual classification with the other dates. First, all predictions are

superimposed and the predicted classes for a single instance of a hemp plant are counted. This process is done for all hemp instances on the image and then, depending on the result, each instance is assigned the resulting class. In each case the class with the most votes is elected:

```
function majority_vote(x, y):
Input x: Predictions from all three dates as array with shape (dates, height, width, classes)
Input y: Date of x to apply majority voting,
Output: Prediction as array with shape (height, width, classes)

n_p = zeros(height, width, classes)

for hemp instance in x[y, :, :, :] do
  m = [0, 0]
  j = 0
  for class 1001 and class 1005 do
    t = sum(x[:, :, :, class], axis=0)
    t[hemp instance = False] = 0
    m[j] = max(t)
  #add instance to class with highest value:
  n_p[:, :, argmax(m)] += hemp instance

#add background class:
b = 1 – sum(n_p, axis=-1)
n_p = concatenate((n_p, b), axis=-1)

return n_p
```

Fig. 11: Basic process of majority voting

If we now look at the visual results in Fig. 17 -

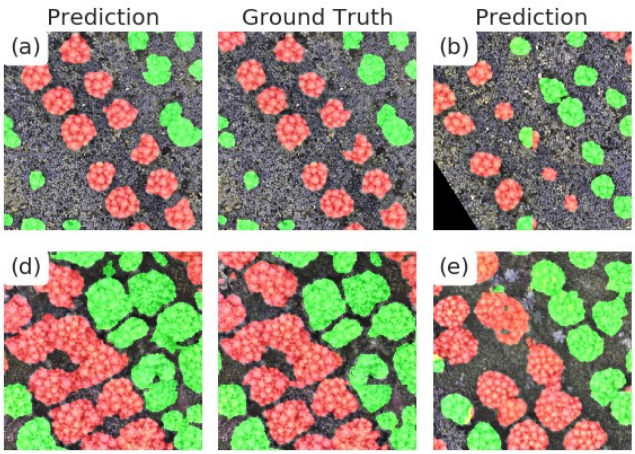

Fig. 19 especially in the first date of 03.07, significant improvements were achieved with regard to correct classification (b), (c), (e). However, at (f) there are still visible errors concerning the classification.

5.3.3 Centre Crop

Furthermore, it can be shown in which regions of the images, based on the respective class, most errors are made. For each image the prediction is compared with the ground truth and the resulting deviation is summed up.

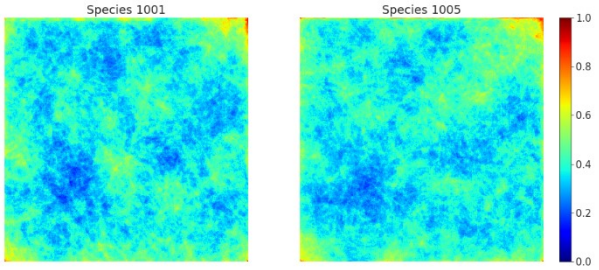

Fig. 12: Superimposed errors regarding classes 1001 and 1005

Considering the error overlays in Fig. 12, most errors occur at the edges of the images. This is probably due to the fact that plants in border- areas are not fully visible and therefore more difficult to classify. To address this problem, predictions are made with an overlap of 64 pixels on each border and then centre-cropped by this overlap before stitching to the map.

Looking at the results in Fig. 21 -

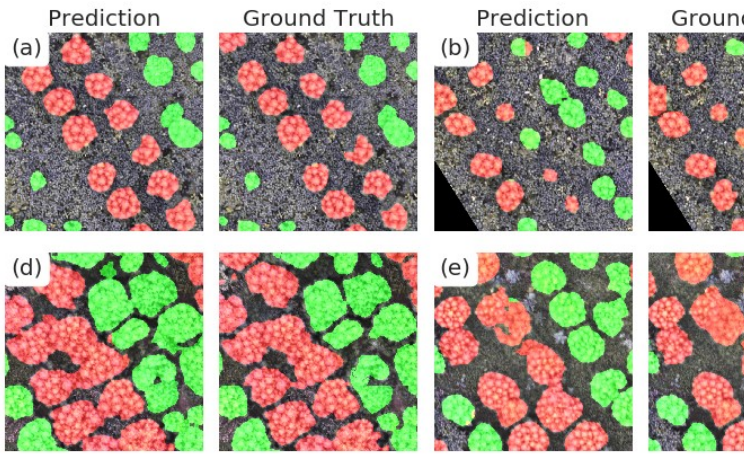

Fig. 23, with the help of the Centre Crop some artefacts have been eliminated (03.07 (b), (e)).

5.3.4 Summary Post Processing

Summarizing it can be said that the results are continuously improving with the help of the applied post processing steps. This is illustrated by the numerical results in Fig. 16, Fig. 20 and Fig. 24 which increase after each step. Thereby an accuracy of over 86% is achieved on the test set which can be seen as a starting point for further implementations and optimizations.

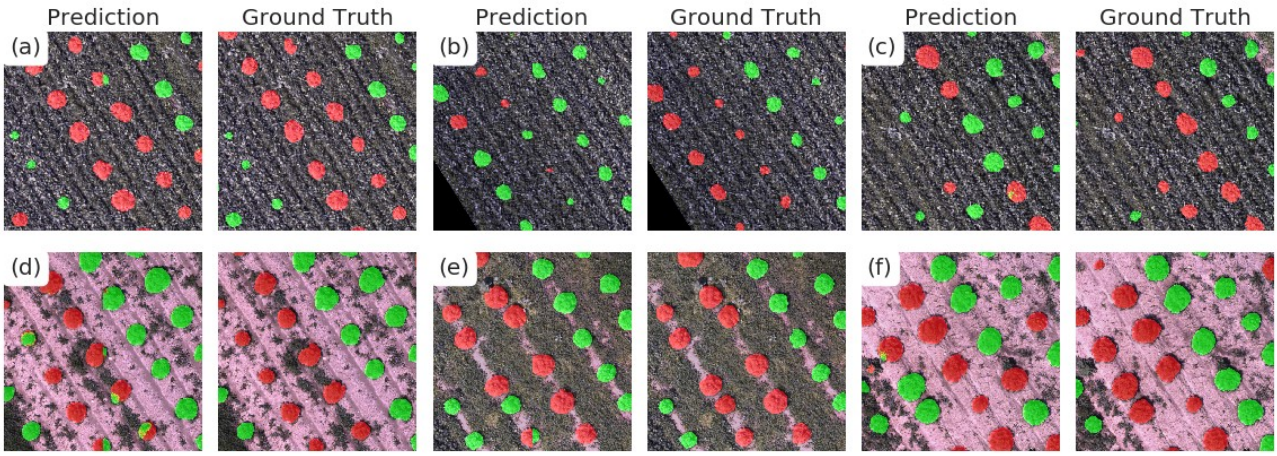

Fig. 13: Date 03.07.2019, no post-processing

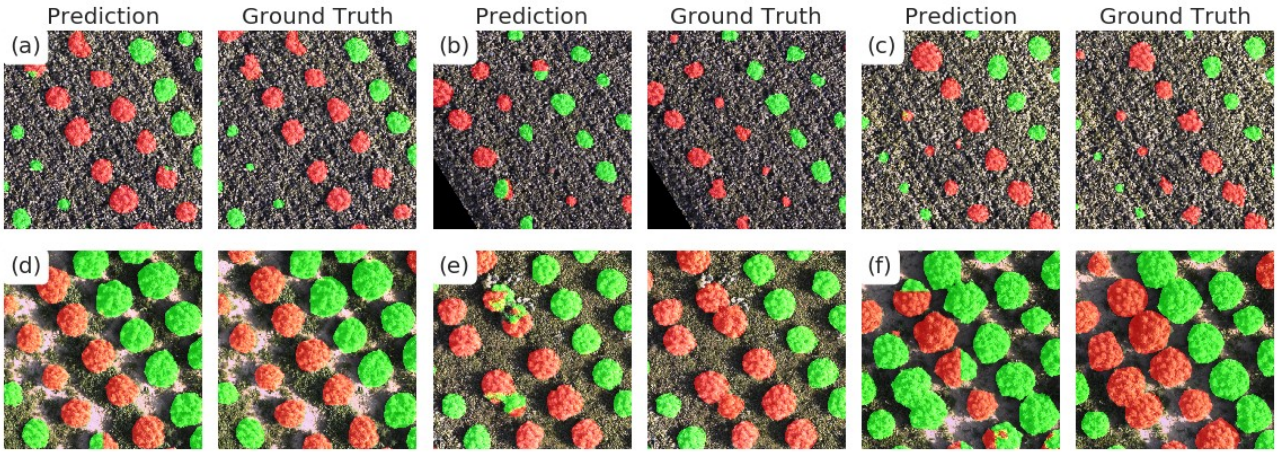

Fig. 14: Date 19.07.2019, no post-processing

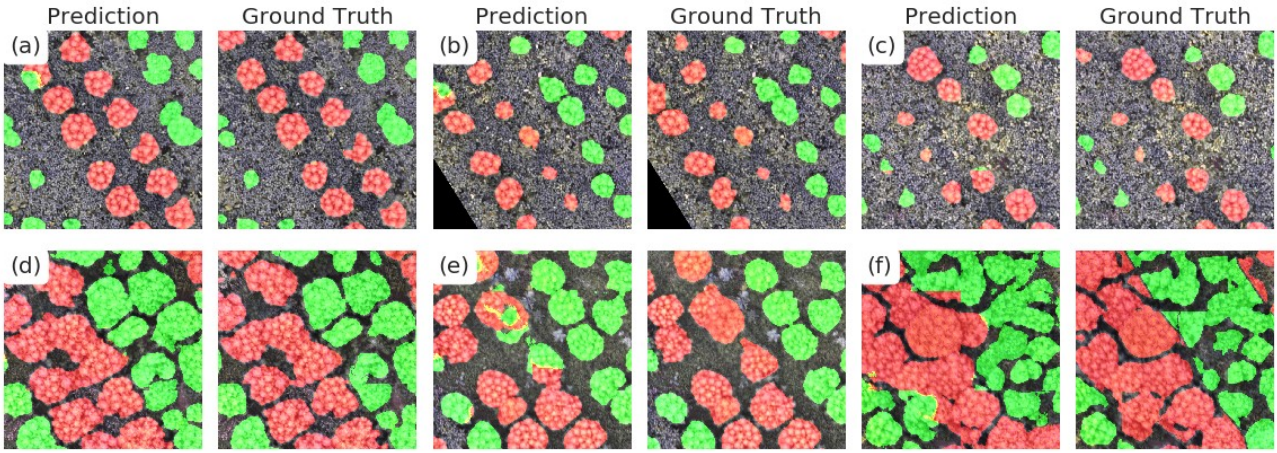

Fig. 15: Date 22.08.2019, no post-processing

Date	Dice Coefficient on stitched Field C (Training / Validation)	Dice Coefficient on stitched Field A (Testing)
03.07	89.43%	81.45%
19.07	93.29%	80.82%
22.08	92.12%	87.72%

Fig. 16: Numerical results with no post-processing

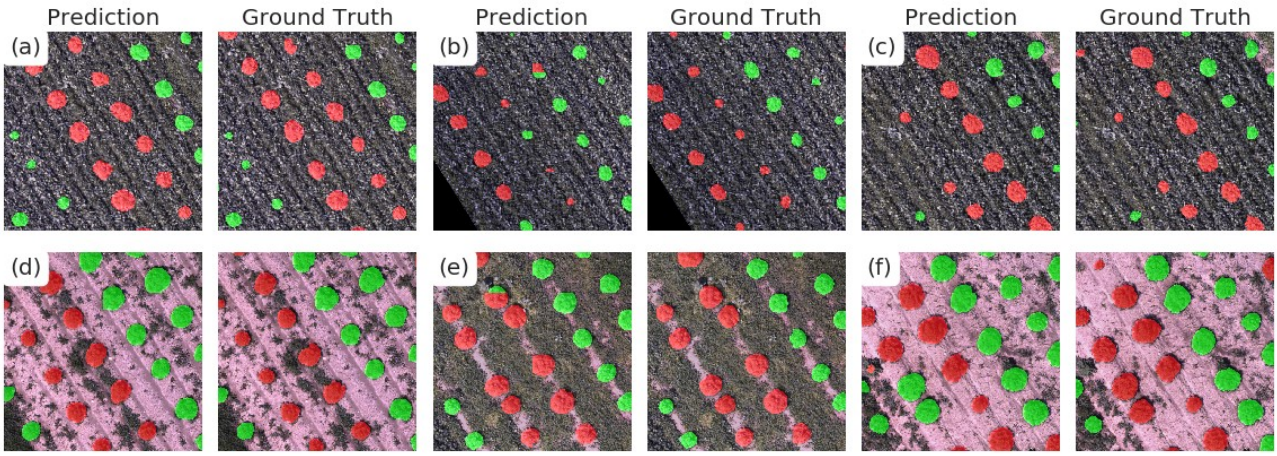

Fig. 17: Date 03.07.2019, majority voting

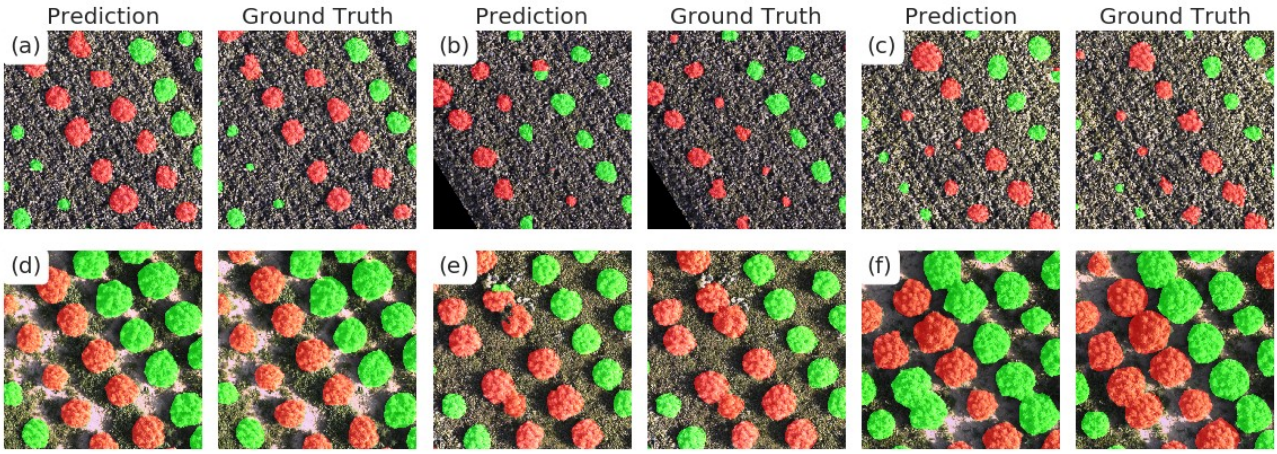

Fig. 18: Date 19.07.2019, majority voting

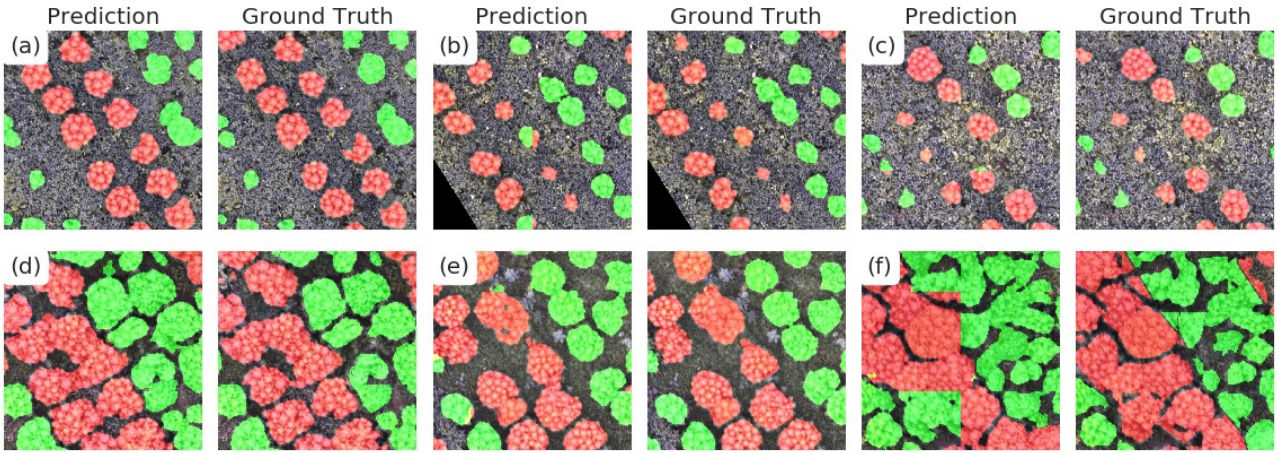

Fig. 19: Date 22.08.2019, majority voting

Date	Dice Coefficient on stitched Field C (Training / Validation)	Dice Coefficient on stitched Field A (Testing)
03.07	93.92%	86.76%
19.07	95.05%	85.24%
22.08	92.92%	88.39%

Fig. 20: Numerical results with majority voting

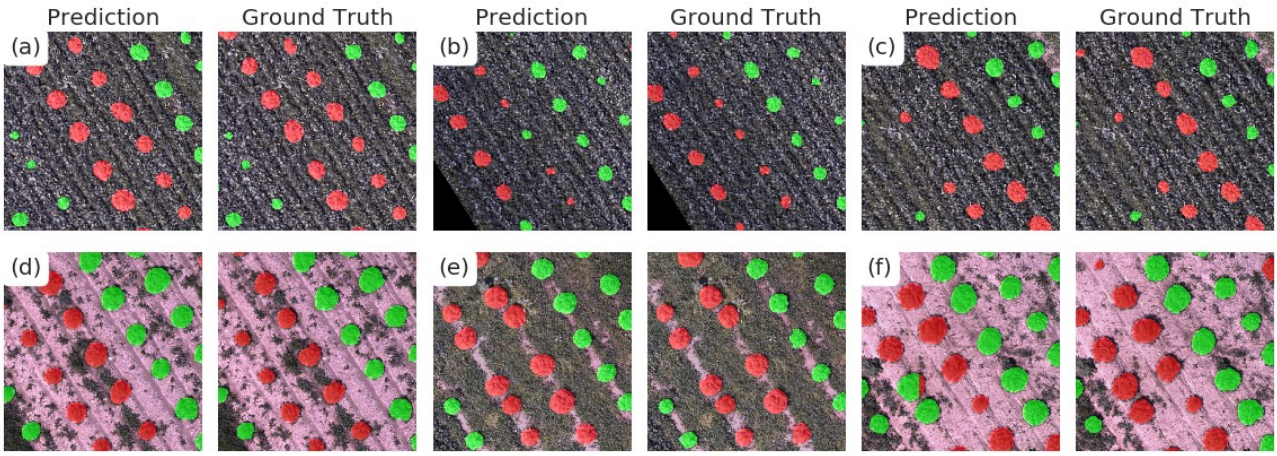

Fig. 21: Date 03.07.2019, majority voting + centre crop

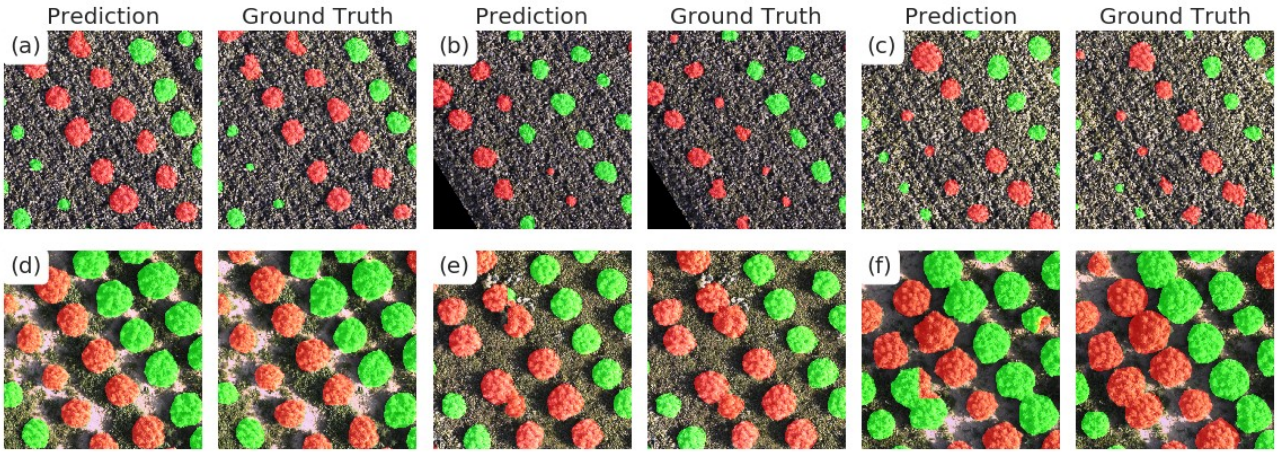

Fig. 22: Date 19.07.2019, majority voting + centre crop

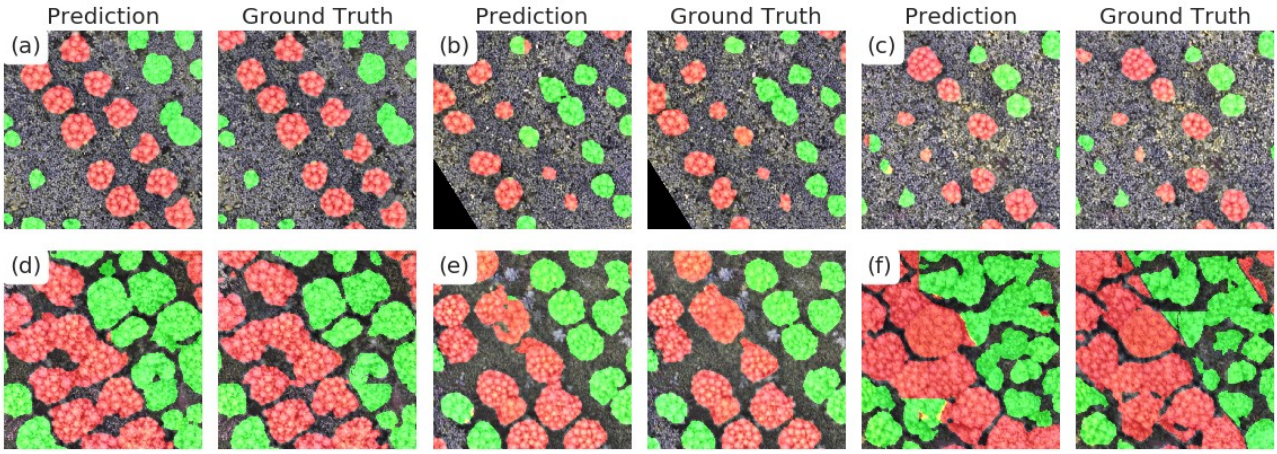

Fig. 23: Date 22.08.2019, majority voting + centre crop

Date	Dice Coefficient on stitched Field C (Training / Validation)	Dice Coefficient on stitched Field A (Testing)
03.07	94.79%	88.36%
19.07	95.76%	86.65%
22.08	93.76%	89.93%

Fig. 24: Numerical Results with majority voting + centre crop

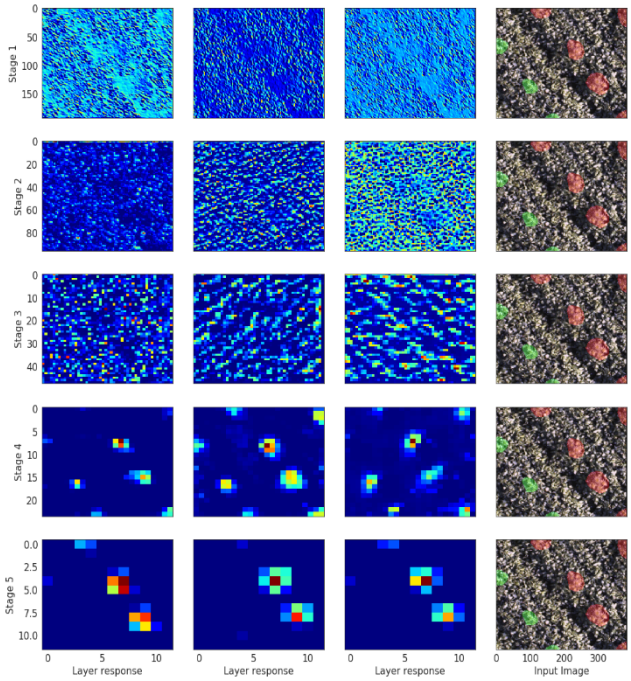

Fig. 25: View of the Layer Activations on an image where the hemp plants are poorly visible.

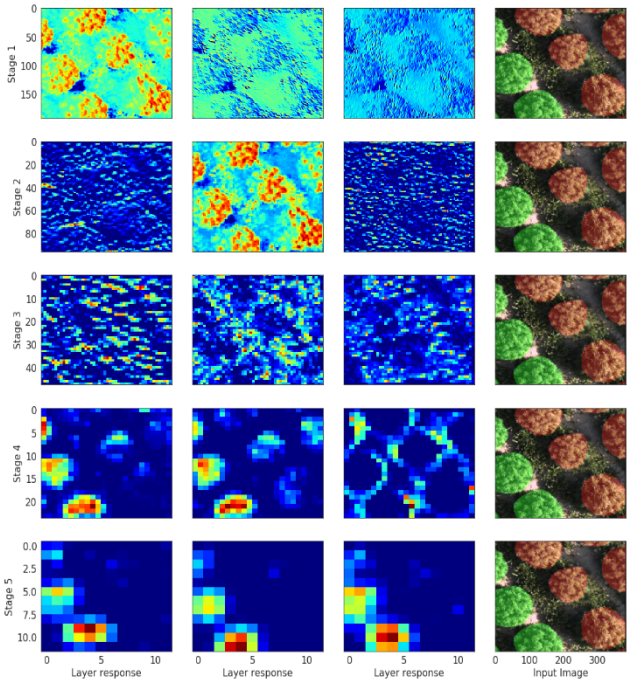

Fig. 26: View of the Layer Activations on an image where the hemp plants are clearly visible.

5.4 Convolutional Feature Maps

As described in Chapter 4.1 the proposed architecture is mainly making use of convolutional operations. To get a better understanding of what the network is learning, individual feature maps can be made visible. One way to achieve this is the visualization of intermediate layers. In order to make the response of certain layers to an input image visible, the network can be "cut apart" at the desired layer and a new “sub-model” can be created. Now the response of the layer can be made visible with a Forward Pass through the network of this “sub-model”.

In Fig. 25 and Fig. 26 the first three columns show outputs of the last layer of each stage from the ResNet encoder. The last column shows the input image which is causing the generated activations.

Looking at Fig. 25 an image with poorly visible hemp plant was fed into the trained model. A hierarchy of learned features can be clearly seen. On the first stages 1-3 mainly edges and corners are recognized, on stages 4-5 more high-level features are visible such as the location of the whole plants. It is visible how the hemp plants are filtered out in stage 4. In Stage 5, class specific features become noticeable as there are some layers which react to class 1001 and some layers react to class 1005.

In contrast considering Fig. 26 an image with clearly visible hemp plants was fed into the trained model. Here, a similar feature hierarchy as in Fig. 25 is visible. It is interesting to see how the layers in Stage 4 react specifically to the regions between the hemp plants.

However, it is not 100% comprehensible which input leads to which decision, but it does give an insight which features are important at which stage.

5.5 Individual Plants Counting

Now individual plants can be counted in the recordings of 03.07 and 19.07. However, the model does not yet provide bounding boxes as an output, but class-specific masks which indicate in which areas hemp occurs and where not. Using these masks, bounding boxes can now be drawn around objects that belong together. But if masks from two different plants are touching each other, the bounding box is drawn around both plants and is therefore not correct. That’s why the plant counting does not work on the recording from 22.08 because a lot of overlapping plants appear in this recording.

To evaluate the accuracy in terms of “is an object detected at the correct position”, a common metric to calculate is the intersection over union (IOU, Fig. 28)

Field	Species	Date	True Positives	False Positives	False Negatives	Precision [%]	Recall [%]
A	1001	03.07	490	78	79	**86.27**	**86.12**
A	1005	03.07	491	104	88	**82.52**	**84.80**
A	1001	19.07	486	110	89	**81.54**	**84.52**
A	1005	19.07	502	105	88	**82.70**	**85.08**
C	1001	03.07	466	34	19	93.20	96.08
C	1005	03.07	456	39	25	92.12	94.80
C	1001	19.07	491	61	25	88.95	95.15
C	1005	19.07	477	61	28	88.66	94.45
A	Both	03.07	1066	97	82	**91.66**	**92.86**
A	Both	19.07	1074	129	91	**89.27**	**92.18**
C	Both	03.07	929	66	37	93.37	96.17
C	Both	19.07	978	112	43	89.72	95.79

Fig. 27: Results Plant Counting

of a predicted bounding box with a ground truth bound box:

$$IOU = \frac{intersection}{union} = \frac{x'y'}{2xy - x'y'}$$

The IOU- metric measures the overlap between two rectangles in the surface. If the rectangles match perfectly an IOU- value of 1.0 results and on the other hand if they do not overlap at all a value of 0.0 is resulting.

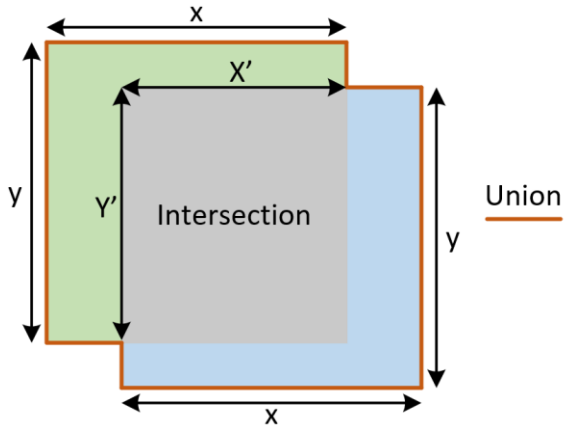

Fig. 28: Representation of the intersection over union metric

Now to measure the accuracy of the plant counting process, for each object combination the resulting IOU value is calculated and stored in a matrix M corresponding the rows to the ground truth and columns to the predicted objects. With this, the following parameters can be calculated:

- True Positive measure; is counted when an IOU $\geq$ 0.5 results. This applies to all rows of the matrix m which have at least one entry greater than 0.5:

$$TP = \sum\sum_{j \in J} M_j \geq 0.5$$

- False Positive measure; is counted when an object is detected but in fact there is none:

$$FP = m - TP$$

- False Negative measure; is counted when an object is not detected but in fact there is one.

$$FN = n - TP$$

- Precision:

$$Precision = \frac{TP}{(TP + FP)}$$

- Recall:

$$Recall = \frac{TP}{(TP + FN)}$$

With:

- M as the IOU- Matrix
- $I = \{1..m\}$ as the indices of the rows of the IOU-Matrix M
- $J = \{1..n\}$ as the indices of the columns of the IOU- Matrix M

According to these calculations the results in Fig. 27 are obtained. Looking at the results, the following can be noticed:

- Mostly a higher recall value then precision, which implies that wrong detections (mostly) are incorrect regarding the class but have not been missed.
- At the later date more plants are recognized. This makes sense because in the first date not all plants are visible.
- The detection of hemp plants regardless of the species results in a higher accuracy, which was already reported in chapter 5.3.

5.6 Volume Calculation

Next, the prediction maps can be used to calculate the volume of the respective species per subfield. A digital surface model (DSM) is available for this purpose. The determined areas [m²] are multiplied by the height [m] of the DSM, resulting in a volume.

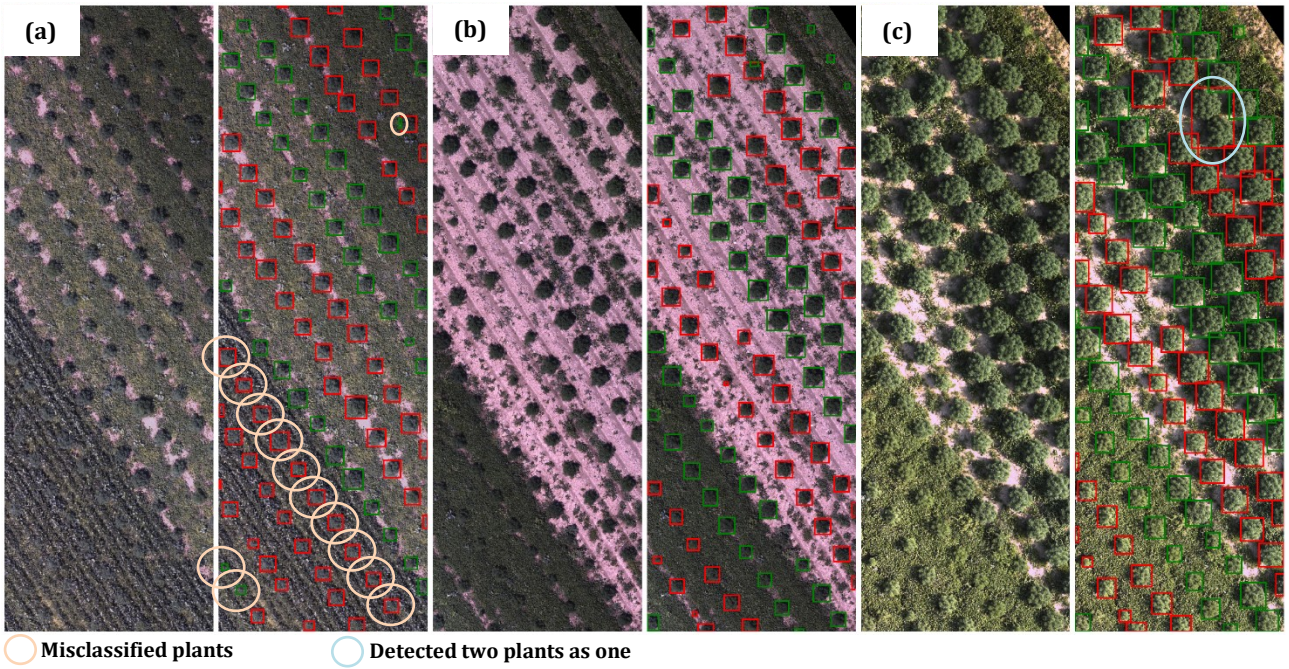

Fig. 29: Individual hemp plant detection results from randomly chosen dates and regions. (**a**) on 03.07.2019, (**b**) on 03.07.2019, (**c**) on 19.07.2019

The DSM was created based on the RGB images and GPS data. To achieve comparable results, the volume is normalized to an area of 100m². As can be seen in Fig. 30 and Fig. 31, the volume increases continuously over time. The fields without subseed deliver thereby in each case the highest yields.

Field A						
	Species 1001			Species 1005		
Date:	03.07	19.07	22.08	03.07	19.07	22.08
Seed						
1-2	1.0	3.5	7.0	1.3	4.0	7.6
3-4	2.9	7.9	19.3	1.5	4.7	10.2
5-6	4.4	12.1	23.7	1.3	4.3	9.8
7-8	3.1	10.0	21.7	2.4	7.5	14.9
9-10	2.4	8.0	22.8	1.6	6.7	18.3
11-12	2.2	12.0	46.5	2.7	13.6	25.9
13-14	1.9	8.5	26.4	2.9	13.7	28.5

Fig. 30: Volume calculation results for Field A in m³/100m²

Field C						
	Species 1001			Species 1005		
Date:	03.07	19.07	22.08	03.07	19.07	22.08
Seed						
1-2	1.8	4.5	8.4	1.5	3.5	7.6
3-4	1.4	3.6	8.3	1.2	3.2	7.1
5-6	2.3	5.8	12.3	1.7	4.8	11.5
7-8	2.8	8.4	26.3	1.0	2.5	6.6
9-10	1.6	4.1	12.1	2.0	6.0	17.1
11-12	2.9	9.9	39.6	2.4	9.5	22.1
13-14	1.0	3.3	19.1	3.4	12.1	36.2

Fig. 31: Volume calculation results for Field C in m³/100m²

6 Discussion

Deep Learning makes it possible to learn representations of data with the help of multi-level abstractions. Choosing the right architecture and data is an important key in successfully creating deep learning applications. The proposed method is based on established algorithms which are already broadly used for segmentation tasks. However, the problem presented here differs from other applications in the following points:

- Limited data available: Since this is a very specific agricultural application, there are few or no public data that can be used for this exact purpose. Therefore, there are no existing data sets with several thousand examples to use for pre-training a model. In addition, it is connected with high expenditure to create own such data sets.
- Visibility: Depending on the subseed, the plants only distinguish themselves very weakly from their background (a plant in front of plants). As an example: When it comes to detecting cars, they usually stand out well from the background and are not in front of many other cars.
- Environmental influences: Environmental variations (e.g., cloudy sky, windy weather, sunlight, shadows) impact the agricultural images in a significant way and hence, make them harder to work with. Similarly, data

samples are also sensitive to imaging angles, field conditions and plant genotypes. Hence, the robustness requirements are significantly higher for the learning models built for agricultural applications.

Under these circumstances it could be shown that the proposed methodology provides a good starting point for segmentation and classification of hemp plants. The following suggestions can be considered for future work:

- Adaptation of the model to achieve instance segmentation. In this way instances of the same class could be separated, e.g. overlapping plants of the same species. For this purpose an extension of the U-Net used here [33] or a Mask-RCNN [34] which allows instance and object detection in one model could be used.
- Further investigations of the multispectral bands to increase the accuracy further.

7 CONCLUSION

In this work, an algorithm was developed which offers a robust method for detection, counting and volume measurement. High-resolution RGB images were acquired by UAV which contain more than 6400 individual hemp plants. A model based on convolutional neural networks and a U-Net architecture was used. The accuracy of the model was improved with post-processing steps. Especially the species recognition in early images could be improved, because in later images the structures of the plants are better visible and therefore the species can be recognized better. Furthermore, possible approaches for plant counting and volume determination were presented and discussed.

8 APPENDIX

8.1 Multispectral Images

[see Notebook Multispektral_Analysis on Github]

As can be seen in the results in chapters 5.3 and 5.5, most errors are not made on the basis of hemp recognition as such, but in the classification of the species. Therefore, the multispectral bands were investigated to find out if they improve the classification.

In order to find out which image bands contribute most to the classification of a hemp species, a data set was created as follows: For each image the hemp plants were extracted according to the ground truth so that only hand plants of both species are visible in the image. Afterwards the pixel median value was extracted for each plant in the image and the corresponding species was noted. Based on this data set, the following three methods were used to determine the relevance of individual bands:

8.1.1 Correlation Matrix

The correlation matrix shows that most bands in the NIR and VIS images correlate strongly. This means that most of the bands do not provide additional information about a single band if they are used as individual bands. For the RGB bands, however, it is visible that they show less correlation to each other. In principle, this observation is understandable since the frequencies of the NIR and VIS bands are very close together whereas the RGB spectra have a larger frequency spacing and therefore each band has a larger information content.

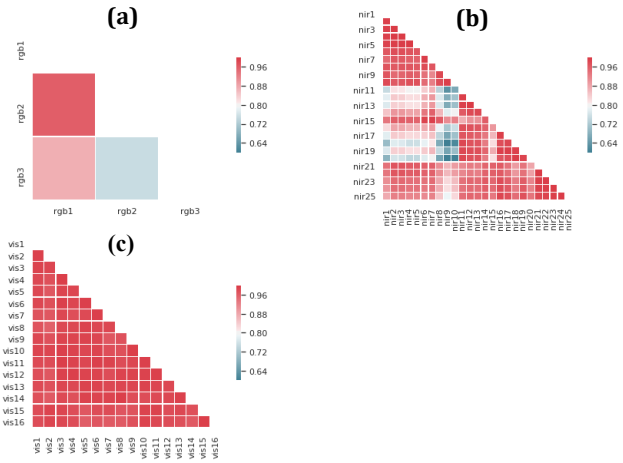

Fig. 32: Correlation Matrix Multispectral Images. RGB in (**a**), NIR in (**b**), VIS in (**c**).

8.1.2 Feature Importance

To determine which features (image bands) lead to the most classification results a decision tree algorithm from the LightGBM library [35] was applied. This algorithm can be used to determine which features are most important for the classification.

Again, it can be seen that the RGB bands are the most important for the classification result.

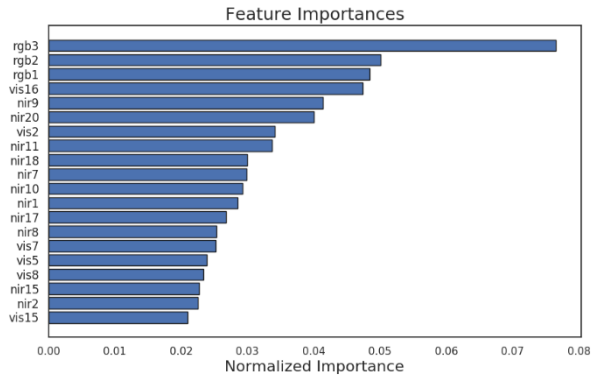

Fig. 33: Feature Importance Multispectral Bands

8.1.3 Principal Component Analysis

One way to avoid the strong correlation described in chapter 8.2.1 is to create linear combinations. A principal component analysis can be used to create "new" bands based on the main existing bands. The goal is to maximize the variance by approximating many statistical variables (the bands in this case) with a smaller number of meaningful linear combinations.

First, the covariance matrix is determined and then the eigenvalues are calculated. Based on the three strongest eigenvalues the image data can be projected along the eigenvector directions which results in the principal components. After reshaping those principal components back to 2D arrays, they can be stacked together resulting in a RGB image or viewed as single grayscale images.

Some interesting results were obtained with the second principal component where stripes appeared. Unfortunately, these do not correlate with the species (Fig. 34).

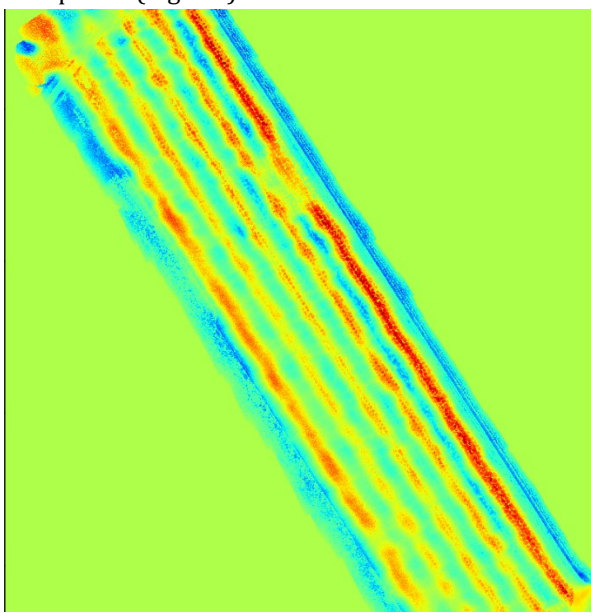

Fig. 34: Principal Component 2 of date 19.07

8.1.4 Summary

Based on the findings, no improvements in classification could be made using the multispectral bands. Also some tests including the principal components did not bring any improvement. This does not mean, of course, that these bands are useless for classification, but no added benefit could be identified in the scope of this work.

9 REFERENCES

[1] D. Fritzsche, "Jedem fünften Schweizer Cannabis-Produzenten droht das Au," 24 07 2019. [Online]. Available: https://www.nzz.ch/zuerich/cannabis-in-der-schweiz-jedem-fuenften-produzenten-droht-das-aus-ld.1496898.

[2] "Cannabidiol, Wikipedia," 20 Februar 2019. [Online]. Available: https://de.wikipedia.org/wiki/Cannabidiol. [Accessed 20 Februar 2019].

[3] S. K. K. I. K. Mochida, "Computer vision-based phenotyping for improvement of plant productivity: a machine learning perspective," *GigaScience,* 2018.

[4] H. S. N. T. A. J. Zhang, "Computer vision and machine learning for robust phenotyping in genome-wide studies," *Scientific Reports,* 2017.

[5] D. B. H. S. S. Ghosal, "An automated soybean multi-stress detection framework using deep convolutional neural networks," *Machine Learning for Cyber-Agricultural Systems,* 2018.

[6] S. J. S. S. A. K. S. A. S. B. G. K. Nagasubramanian, "Hyperspectral band selection using genetic algorithm and support vector machines for early identification of charcoal rot disease in soybean stems," *Plant Methods.*

[7] S. J. A. K. S. A. S. B. G. K. Nagasubramanian, "Explaining hyperspectral imaging based plant disease identification identification: 3d cnn and saliency maps," *Arxiv.*

[8] D. B. A. K. S. B. G. A. S. a. S. S. S. Ghosal, "An explainable deep machine vision framework for plant stress phenotyping," *Proceedings of the National Acadamy of Sciences of the United States of America,* 2018.

[9] U. K. R. S. N. W. Guo, "Illumination invariant segmentation of vegetation for time series wheat images based on decision tree model," *Computers and Electronics in Agriculture,* 2013.

[10] R. K. J. P. R. S. C. S. P. Lottes, "UAV-based crop and weed classification for smart farming," *Proceedings of the 2017 IEEE International Conference on Robotics and Automation,* 2017.

[11] D. M. F. G. G. d. S. H. P. a. M. F. A. dos Santos Ferreira, "Weed detection in soybean crops using ConvNets," *Computers and Electronics in Agriculture,* 2017.

[12] G. J. R. F. M. Louargant, "Unsupervised classification algorithm for early weed detection in row-crops by combining spatial and spectral information," *Remote Sensing,* 2018.

[13] P. R. D. W. H. H. S. Varela, "Early-season stand count determination in Corn via integration of imagery from unmanned aerial systems (UAS) and supervised learning techniques," *Remote Sensing,* 2018.

[14] Y. F. D. T. Y. Mu, "Characterization of peach tree crown by using high-resolution images froman unmanned aerial vehicle," *Horticulture Rese,* 2018.

[15] W. G. Y. Y. a. S. N. K. Yamamoto, "On plant detection of intact tomato fruits using image analysis and machine learning methods," *Sensors,* 2014.

[16] T. F. S. N. W. Guo, "Automated characterization of flowering dynamics in rice using field-acquired timeseries RGB images," *Plant Methods,* 2015.

[17] Z. G. F. D. B. U. T. P. C. M. I. Sa, "Deepfruits: a fruit detection system using deep neural networks," *Sensors,* 2016.

[18] X. J. H. L. S. Madec, "Ear density estimation from high resolution RGB imagery using deep learning technique," *Agricultural and ForestMeteorology,* 2019.

[19] J. P. C. H. L. S. J. M. M. M. Hasan, "Detection and analysis of wheat spikes using convolutional neural networks," *Plant Methods,* 2018.

[20] L. D. L. L. X. Xiong, "Panicle-SEG: a robust image segmentation method for rice panicles in the field based on deep learning and superpixel optimization," *Plant Methods,* 2017.

[21] S. Ghosal, B. Zheng, C. S. Chapman, B. A. Potgieter, R. D. Jordan, X. Wang, A. K. Singh, A. Singh, M. Hirafuji, S. Ninomiya, B. Ganapathysubramanian, S. Sarkar and W. Guo, "A Weakly Supervised Deep Learning Framework for Sorghum Head Detection and Counting," *Plant Phenomics,* vol. 2019, p. 14, 2019.

[22] B. Neupane, T. Horanont and N. D. Hung, "Deep learning based banana plant detection and counting using high-resolution red-green-blue," *Plos One,* p. 22, 2019.

[23] Y. T. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, "Focal Loss for Dense Object Detection," *arXiv,* p. 10, 2018.

[24] M. Houmi, B. Mohamadi and T. Balz, "A HYPERSPECTRAL BASED METHOD TO DETECT CANNABIS PLANTATION IN INACCESSIBLE AREAS," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Vols. XLII-3, p. 5, 2018.

[25] "QGIS, Ein freies Open-Source-Geographisches-Informationssystem," 20 02 2020. [Online]. Available: https://www.qgis.org/de/site/. [Accessed 20 02 2020].

[26] O. Ronneberger , P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," p. 8, 2015.

[27] E. S. T. D. Jonathan Long, "Fully Convolutional Networks for Semantic Segmentation," *arXiv.org,* 2014.

[28] M. D. D. V. A. R. Y. B. Simon Jégou, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," *arXiv.org* , 2016.

[29] X. Z. S. R. J. S. Kaiming He, "Deep Residual Learning for Image Recognition," *arXiv.org,* 2015.

[30] J. D. H. S. J. K. S. S. S. M. Z. H. A. K. A. K. M. B. A. C. B. L. F.-F. Olga Russakovsky, "ImageNet Large Scale Visual Recognition Challenge," *arXiv.org,* 2014.

[31] Google Brain, "Tensorflow," 10 02 2020. [Online]. Available:

https://www.tensorflow.org/api_docs/python/tf/image. [Accessed 19 02 2020].

[32] Google Brain, "TensorFlow," 2020. [Online]. Available: https://www.tensorflow.org/. [Accessed 19 February 2020].

[33] R. U. Min Bai, "Deep Watershed Transform for Instance Segmentation," *arXiv.org,* 2016.

[34] G. G. P. D. R. G. Kaiming He, "Mask R-CNN," *arXiv.org,* 2018.

[35] Microsoft, "LightGBM's documentation," Microsoft, 25 02 2020. [Online]. Available: https://lightgbm.readthedocs.io/en/latest/. [Accessed 25 02 2020].