Automated Detection and Segmentation of Grain Spikes in Greenhouse Images Using Shallow and Deep Learning Neural Networks: A Comparison of Six Methods

Sajid Ullah (✉ sajid.ullah@ceitec.muni.cz)

Masaryk University

Michael Henke

Masaryk University

Narendra Narisetti

Institute of Plant Genetics and Crop Plant Research

Jan Hejatko

Masaryk University

Evgeny Gladilin

Institute of Plant Genetics and Crop Plant Research

Research Article

Keywords: high-throughput plant image analysis, spike detection, spike segmentation, deep learning, automated plant phenotyping

Posted Date: March 2nd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-252740/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Automated detection and segmentation of grain spikes in greenhouse images using shallow and deep learning neural networks: a comparison of six methods

Sajid Ullah^{1,*}, Michael Henke¹, Narendra Narisetti², Jan Hejatko¹, and Evgeny Gladilin^{2,**}

¹Plant Sciences Core Facility, CEITEC-Central European Institute of Technology, Masaryk University

²Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Gatersleben, Germany

*sajid.ullah@ceitec.muni.cz

**gladilin@ipk-gatersleben.de

ABSTRACT

Image-based plant phenotyping is the major approach to quantitative assessment of important plant properties. For automated analysis of a large amount of image data from high-throughput greenhouse measurements, efficient techniques for image segmentation are required. However, conventional approaches to whole plant and plant organ segmentation are hampered by high variability of plant and background illumination, and naturally occurring changes in geometry and colors of growing plants. Consequently, application of advanced machine learning techniques for automated image segmentation is required. Here, we investigate six advanced neural network (NN) methods for detection and segmentation of grain spikes in RGB images including three detection deep NNs (SSD, Faster-RCNN, YOLOv3/v4), two deep (U-Net, DeepLabv3+) and one shallow segmentation NNs. Our experimental results show superior performance of deep learning NNs that achieve in average more than 90% accuracy by detection and segmentation of wheat as well as barley and rye spikes. However, different methods demonstrate different performance on matured, emergent and occluded spikes. In addition to comprehensive comparison of six NN methods, a GUI-based tool (SpikeApp) provided with this work demonstrates the application of detection and segmentation NNs to fully automated spike phenotyping. Further improvements of evaluated NN approaches are discussed.

Keywords: high-throughput plant image analysis, spike detection, spike segmentation, deep learning, automated plant phenotyping

Introduction

Grain plants such as wheat, barley and rye are foremost consumed cereal crops worldwide with wheat having a steady growth of demand of 15%.¹ For 2050, an additional increase of 90 million metric tonnes was forecast.² With prognosticated increase of demand on grain crops in the next decades, more efficient and cost-effective approaches to yield enhancement and breeding are required. To assess effects of genetic perturbations and environmental conditions on biological plant traits image-based high-throughput plant phenotyping in a controlled greenhouse environment is frequently performed. Derivation of reliable quantitative traits (QTs) such as morphological and developmental features became a method of choice by investigation of effects of biotic and abiotic factors on plant growth and grain yield.³ However, due to high variability of optical plant appearance image-based phenotyping turned out to render a non-trivial task, which represents one of the major bottlenecks of quantitative plant science.^{4,5} In addition to assessment of the overall plant biomass and structure, detection and quantification of plant organs, such as wheat ears and spikes, is of particular interest for biologists and breeders. Density of spikes per unit field area is one of the key yield descriptors, while size, color and number of grains provide valuable insights into a more detailed assessment of wheat development and quality of production. In context of spike image analysis, two major tasks are of particular interest: (i) detection/counting and (ii) pixel-wise segmentation of spikes complemented by their subsequent phenotyping, see examples in Fig. 1. These closely related tasks were already addressed in a number of previous works dealing with plant organ detection and segmentation. For example, *Grillo et al.* applied image analysis techniques to identify wheat landraces based on glume phenotypes by statistical analysis of morpho-colorimetric descriptors.⁶ *Bi et al.* designed different architectures of 3-layer neural network based on number of hidden layer nodes to classify four wheat varieties in sole-spike image to extract spike traits such as the awn number, the average awn length and the spike length.⁷ *Misra et al.* presented

SpikeSegNet which performs spike detection with two cascaded feature networks: local patch extraction and global mask refinement network.⁸ *Hasan et al.* achieved the spike detection and counting with R-CNN obtaining a F1 score of 0.95 on 20 wheat field images having an average of 70-80 spikes per image.⁵ *Tan et al.* applied support vector machine (SVM) and k-nearest neighborhood for wheat spike recognition on pre-segmented spike regions and super-pixels that were generated by Simple Linear Iterative Clustering. *Alharbi et al.*⁹ detected wheat ears by transforming the raw plant images using color index of vegetation extraction (CIVE) and performed clustering of pixel features extracted in CIVE domain using k-means. *Pound et al.* implemented deep neural network (DNN) for identification of spikelets and their counting.¹⁰ *Amara et al.* *Qiongyan et al.* presented a shallow artificial neural network for segmentation of wheat spikes, which showed a satisfactory performance on wheat cultivars exhibiting spikes growing on the top of the plant ('top spikes').¹¹ However, in our previous work,¹² we have found out that such a shallow ANN is rather restricted to detection of similar top spikes and does not perform that good for more bushy European wheat cultivars that exhibit spikes in the middle of the plant surrounded and partially overlaid by leaves. Improvements introduced to the shallow ANN architecture such as Frangi line filters could enhance the final segmentation results, however, this framework still requires substantial efforts for manual adjustment by application to the new image data. Most of the above mentioned works are typically restricted to a particular subset of image data, and rarely provide source code or deployed tools for reproducing the results and routine application.

With the success of AlexNet,¹³ significantly more robust and accurate results of image segmentation were achieved in widely automated manner as compared to traditional classification techniques based on a predefined set of features. The top performance of DNNs on benchmark data set i.e. VOC2007-12 and MS COCO is attributed to automated feature extraction of classifier and pixel-wise segmentation.^{14,15} Meanwhile, a large number of DNN architectures was reported for the frequently demanded tasks of pattern detection and image segmentation. However, studies demonstrating performance of different DNNs in application to plant and, in particular, plant organ detection/segmentation are relatively rare.

Consequently, the main objective of the present study was to investigate and compare the performance of different deep learning frameworks for the task of spike detection and segmentation. Once detected or segmented, spike regions can be quantitatively characterized in terms of various color, shape features that are essential to plant biologists and breeders. The automated detection of spikes is known to be dependent on several factors including spike size, shape, texture and location within the plant that are in turn varying from one grain type to another. Accordingly, we investigate the effects of spike optical appearance and location on performance of DNN models. To overcome the limitations and low-applicability of previous works, here we performed comparative investigation of three detection DNNs including Single Shot Multibox Detector (SSD), Faster-RCNN, YOLOv3/v4, as well as two segmentation DNNs (U-Net, DeepLabv3+) and one conventional shallow ANN. Our work gives comprehensive insights in quantitative performance of six different methods for detection and segmentation of different spike phenotypes in wheat, barley and rye. Furthermore, we present a user-friendly GUI-based tool (SpikeApp) which demonstrates automated spike detection, segmentation and phenotyping using three pre-trained neural network models including U-Net, YOLOv3 and shallow ANN.

Methods

Image acquisition

Wheat plant images were acquired from high-throughput greenhouse phenotyping system of Photon System Instruments (PSI) (<https://psi.cz/>). Twenty-two cultivars of Central European wheat were imaged in vegetative and reproductive stage, taken in PSI photo chamber. Out of twenty-two cultivars, nineteen were selected for spike detection and segmentation task. An overview of wheat cultivars analyzed in this work including the number of RGB visible light images of each cultivar is summarized in Table 1. The plant images for the experiment were captured in the side view from two rotational angles (0° and 90°). Images were taken in the same resolution 2560x2976 using the uniform blue background.

Data set preparation

The deep convolution neural network (DNN) used for spike detection were trained on original images of the size 2560x2976 stored in the PNG format. The training data set was reiterated for reduced resolution of 800x600. The multi-resolution testing of DNNs was necessary to ensure whether the DNN can preserve high frequency information of spike boundary in lower resolution. The annotations for the spike detection were done with LabelImg¹⁶ by drawing a bounding box around each spike and subsequently saved as *.xml file as required for Faster-RCNN and SSD. For YOLO, the annotations were converted to *.json file. The spike labeling for segmentation was accomplished by GIMP image processing software¹⁷ with Free Select tool and Bucket Fill. Labeled structures were saved as grayscale images. The segmentation is regarded as binary pixel-wise labeling with spike region having the value of 1 and non-spike region the value 0. The training set consists of 234 wheat images from nineteen cultivars. 219 wheat plants were imaged through its life cycle to point the spikes were mature for harvesting. Out of the 234 images, 33% of the plant images were taken from two direction side views (0° and 90°). The testing set comprises 58 images including 8 images that contain spikes occluded by leaves or in some cases stem of the plant.

The training set contains 203 images of Green Spike and Green Canopy (GSGC), 27 images of Yellow Spike and Yellow Canopy (YSYC) and 4 negative training (no spikes) images. These training images having fewer leaves in plant compared to the high yielding Central European wheat plant (in generalization test) in which the challenge is to detect spikes that exhibit colors similar to leaves. The output of spike detection represents a list of bounding boxes with class (spike) probability. The spike segmentation output is labeled with spike or background (non-spike) region.

Intersection over Union (IoU) is computed as the area of the intersection divided by the area of the union of a predicted bounding box B_p and a ground-truth box B_{gt} . Mean average precision (mAP) is computed from area under Precision-Recall curve of different recall point and then taking average of all classes.

The bounding box regression from spike detection DNN is classified as

- **True positive (TP)**

The output frame is assigned true positive when the bounding box contain spike when IoU is 0.5 or more and is dependent on AP value used for evaluation. However, in Microsoft Common Object in Context (COCO) detection evaluation metrics, the IoU value of 0.75 or above is suggested, see discussed in section

- **False positive (FP)**

False positive is either background classified incorrectly as spike or several regions classified as spike (multiple bounding box overlapping on single spike).

- **False negative (FN)**

The output frame from DNN in which the spike region is incorrectly classified as a background.

- **True negative (TN)**

In this classification problem, the background is classified as true background.

Precision (P), Recall (R), Accuracy (A) and F1 measures are calculated based on standard detection benchmarks such as PASCAL VOC and COCO.

Positive prediction value /Precision: the number of true spike frames correctly classified as spike.

$$P = \frac{TP}{TP + FP} \quad (1)$$

True positive rate /Recall: the number of spikes in test image that has been localized with bounding box ($\text{IoU} \geq 0.5$)

$$R = \frac{TP}{TP + FN} \quad (2)$$

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

The model robustness is quantified by calculating harmonic mean of Precision and Recall

$$F1 = 2 \frac{P * R}{P + R} \quad (4)$$

We have evaluated our data set with commonly used metrics for object detection such as PASCAL VOC and COCO detection measures. The mAP is used to evaluate the localization and class confidence of spike as in Equation 5

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5)$$

In PASCAL VOC 2007, the average precision, AP is calculated at single IoU value whereas on COCO evaluation, that is more stringent evaluation measure than PASCAL VOC, AP is calculated at ten different IoU thresholds (0.5:0.05:0.95) while the final mAP of DNN is averaged over the 10 IoU threshold values. The mean of the average precision is computed on both classes: spike and background. The binary output from segmentation task is evaluated by the Dice coefficient score. A binary mask of prediction is the output with zeros for non-spike and ones in cases for spike pixels.. The F1 score for segmentation in contrast to the spike detection is done at the pixel level. We also evaluated the test set with IoU/Jaccard index. Both the evaluation measures for segmentation are positively correlated.

Spike detection DNN models

In this section describes DNN models for spike detection starting from deep to deeper neural network: SSD, Faster-RCNN and YOLOv3/v4.

Single shot multibox detector

SSD is feed-forward convolutional network that makes multiple predictions for the bounding box of spikes crossover different scales.¹⁸ SSD has the characteristics of generating region proposal like YOLOv3. The single stage detector divides the image into grid cells, and each cell has a likelihood of spike being located in it. In case of multiple objects in a grid, the SSD in-training process deploys a pre-defined aspect ratio of anchor boxes and produces a score for each object in each box. It extracts the feature maps at different locations of input map in sliding window fashion. At lower resolution, the feature map is suited to extract the feature of large objects and at higher resolution feature map extracts small objects as in our case, spikes. The backbone is cascaded with fully connected layers that output the classes and bounding box locations in predicted object regions. The grid that has multiple anchors box overlapped over the candidate object. In that case, anchor box that has the largest overlapped ($IoU > 0.5$) over the ground truth is picked out as object. The localization loss function is the mismatch between the ground truth and predicted boundary box. Only positive match is penalized, while negative match is ignored while computing loss. The total loss in prediction is the sum over validation/testing positive matches of localization and confidence loss. The multiple predictions over the single spike (overlapping boxes) during inference are mitigated by deploying non-maximal suppression, which picks the predicted spike frame with the highest probability.

Faster-RCNN

Selective search¹⁹ has been used successfully for region proposals generation. On the other hand Faster RCNN deploys a small DNN for feature extraction for region proposal network (RPN). In recent years, RPN has been used prior to main object detector, which produces candidate objects with objectness score for the object detection and has been implemented successfully to many publicly available data sets.²⁰ In this work we have implemented a two-stage cascaded detection Faster-RCNN framework.²¹ In the first stage, multiple objects are proposed prior to extracting features for the main detector. These proposals are spike or background. Its deployment improves the classifier accuracy, but this improvement come at the expense of an increase in computational resources. The RPN is implemented as fully convolutional network and works as mini-network trained end-to-end with main object detector. RPN and region-based object detection DNN share the same convolution feature. The computational speed of RPN compared to selective search space is 10 ms to 0.2 sec per image.²¹ The features in convolution layers are shared in RPNs and DNN object detectors. The RPNs are fed directly to the DNN object detector. The DNN initializes the weights in RPNs with zero-mean Gaussian distribution and standard deviation 0.01. In the first stage processing, 100 spike proposals were extracted for features and regression. Each anchor/proposal generated have either positive anchor label or negative anchor label. Positive label is assigned to those one which has the highest IoU with ground truth or IoU overlap higher than 0.7, and the negative anchor label is one in which IoU is lower than 0.3. The RPN computes the features map with six different anchor size differentiated by the aspect ratio which is initialize at the beginning of the training process. Therefore, each regressor is responsible for extracting weight in separate spatial size ($n \times n$) on feature map. The loss (softmax loss for classification and L1 loss for regression) in RPN is calculated in mini-batch to mitigate the bias towards background space as it is dominated by negative anchors. The negative anchors are background samples compared to the foreground which occupy less spatial size. This is done to keep the the classifier not bias towards the over-sampled anchors from background. In second stage of Faster-RCNN, exponential decay is used as learning parameter for training images. The features computed in RPN are passed through ROI pooling layer and turned into feature vector for fully connected layer in main detector. Finally, a softmax layer produced a binary output assigned with set of class probabilities and regressor computes bounding box with accurate coordinates. Typical curves of in-training loss and average precision during the training process of Faster-RCNN are depicted in Fig. 2(a).

YOLOv3 and YOLOv4

The YOLOv3 and its v4 variant differ from region proposed networks by the selection of initial proposal for the feature map extraction. YOLOv3 divides in-training image into fixed number of $S \times S$ grid. The class label is predicted for single object in grid cell.²² For each grid cell, the fully connected layers output bounding box and confidence score computed in single forward pass from conditional class probabilities. The objectness score for bounding box was computed using logistic regression. The YOLOv3 variant is faster for real time object detectors by dividing the image into fixed grid. As backbone in YOLOv3, we implemented Darknet53 while for YOLOv4 we took CSPDarknet53. In YOLOv4, Mish activation function is used on output convolution layer in feature extractor and detector.²³ The training loss for class prediction used is binary cross-entropy while sum squared error is used for the calculating the loss. The network has cascaded 3×3 and 1×1 convolutional layers. The skip connection which bypass certain layers resulting in uninterrupted gradient flow. The size of layers skipping is more in Darknet53 than its predecessor Darknet19. The shortcuts connection skips the detection layer that do not decrease the loss

on those layers. The spike prediction is done across three scales in detection layers. The bounding boxes are predicted with dimension cluster: the output, 4D tensor prediction of bounding box consists of four coordinates: t_x , t_y , t_w and t_h . Logistic regression is used to compute the objectness score for every bounding box. If overlap between the predicted bounding box and ground truth is 0.5, the class probability of bounding box has confidence of 1. Logistic classifier is deployed at the prediction layer for classification.

The efficient use of defining objects in individual cell makes it competitive edge over other state-of-the-arts DNNs for instance ResNet101, ResNet152 particular for real time application⁽²⁴⁾. The training process of YOLOv3 is depicted in Fig. 2(b). The network was trained on image size of 2560x2976. The training process took nine hours. One of the improvement of YOLOv4 over YOLOv3 is introduction of mosaic image enhancement. The image augmentation of CutOut, MixUp and CutMix were implemented. The loss function used in training of the YOLOv4 include classification loss (L_{class}), confidence loss ($L_{confidence}$) and bounding box position loss (L_{IoU}).²³

$$Net\ loss = L_{class} + L_{confidence} + L_{IoU} \quad (6)$$

Spike segmentation models

The section gives description of spike segmentation NNs including two DNNs (U-Net, DeepLabv3+) and a shallow ANN.

Shallow artificial neural network

The shallow artificial neural network (ANN) approach from¹¹ with extensions introduced in¹² was retrained with ground truth segmentation data for leaf and spike patterns from the training set. Texture law energy well known from several previous works^{7,25,26} was used in this approach as the main feature. As pre-processing step, grayscale image is converted to wavelet discrete wavelet transform (DWT) using the Haar basis function. The DWT is used as input to shallow ANN. In the first feature extraction step, nine 3x3 convolution mask of size $2n + 1$ are convolved with original image I . The convolutional equation for this step is given by Equation 7:

$$F(i, j) = (A * I)(i, j) \sum_{p=-a}^a \sum_{l=-a}^a I(i+p, j+l) \quad (7)$$

In the second step, the mean deviation around pixel is computed by macro-windowing operation of size $(2n + 1)(2n + 1)$ on neighborhood of every pixel and computed with Equation 8:

$$E(i, j) = \frac{1}{(2n + 1)} \sum_{p=i-n}^{i+n} \sum_{l=j-n}^{j+n} |F(p, l)| \quad (8)$$

Finally, the boundaries obtained from ANN are filtered using multi-scale Frangi-filter to eliminate noisy edges as described in.¹²

U-Net

In this work, the U-Net architecture from²⁷ was extended to process RGB spike images. U-Net consists of down sampling path in which the feature map is doubled in encoder block while image size is reduced by half. Each of five blocks of contracting path consist of consecutive 3x3 *conv* layer and followed by *Maxpool* layer. The plateau block has also a pair of consecutive *conv* layer without *Maxpool* layer. The layers in expansive path is concatenated with corresponding layer for feature map in contracting path which makes the prediction boundary of the object more accurate. In expansive path, the size of image is restored in each *transposed conv* block. The feature map from *conv* layer in expansive path is batch normalized. The final layer is 1x1 *conv* layer with 1 filter which produces the output binary pixels. The U-Net is fully convolutional network without any dense layers. The training process was performed with Relu as an activation function. In the output of the U-Net prediction, the value of 0 is assigned to background and value of 1 to spike pixels resulting in binary pixel-wise image segmentation. U-Net model was optimized by Adam optimizer²⁸ with variable learning rate scheduling decreasing with each epoch from $3E - 3$ to $1E - 5$. In order to enable training the U-Net model on the original image resolution including important high-frequency information, original images were cropped into masks of 256x256 size. Using of full-size original images would not be possible due to limitations of our GPU resources. Since spikes occupy only very small image regions, usage of masks helped to overcome limitations by processing of the full-size images while preserving high-frequency information. To mitigate the class imbalance issue and to remove the frames that solely have blue background and kept the ratio of spike vs non spike (frame) regions as 1:1.

DeepLabv3+

DeepLabv3+ is a state-of-the-art segmentation model that has shown a relatively high mIoU of 0.89 on PASCAL VOC 2012.²⁹ The performance improvement is particular attributed to Atrous Spatial Pyramid Pooling (ASPP) module which obtains contextual information on multi-scale at several atrous convolution rates. In DeepLabv3+, atrous convolution is an integrated part of the network backbone.³⁰ employed atrous convolution to mitigate the reduction of spatial resolution of feature responses. Input images are processed using network backbone. The output is elicited from each location i and filter weight w . The atrous convolution is processed over the feature map. The notation for atrous convolution signal is similar to one used in.³¹ For location i and filter weight w . When atrous convolution is applied over feature map x , the output y defined by the Equation 9

$$y[i] = \sum_{k=1}^K [i + r.k]w[k] \quad (9)$$

where r denotes the rate with which the input signal is sampled. The feature response is control by atrous convolution. The output stride is defined as the ratio of input spatial resolution to output spatial resolution of feature map.³¹ A large-range link is established between the network backbone and multiple-scale feature extraction modules: ASPP and dense prediction cell (DPC). The depth-wise separable convolution operates convolution on each channel followed by point-wise convolution (1x1) which superimposes feature signal from the individual channel.

In decoder part, the features are bilinear upsampled. The output of which is convolved with 1x1 convolution and then concatenate with low-level features. Another 3x3 convolution is operated on feature map followed by bilinear upsampling and the output is binary semantic labels. Here, we modified and implemented publicly available DeepLabv3+³² for training and evaluation on our spike image data set. In this study, DeepLabv3+ was trained for 20k epoch with batch size of 6. The polynomial learning rate was used with weight decay of $1E - 4$. The output stride for spatial convolution was kept at 16. The learning rate of the model was $2E - 3$ to $1E - 5$ with weight decay of $2E - 4$ and momentum of 0.90.

By training of U-Net and DeepLabv3+, conventional augmentation techniques including rotation [-30 30], horizontal flip, and image brightness change [0.5 1.5] were adopted. The ratio of the augmented images has the same proportion of GSGC:YSYC and also the non-spike images as previously used in our training set for detection DNNs.

Evaluation of spike detection models

The DNNs deployed in this work are evaluated by mAP which is computed as a weighted mean of precision at different threshold values of recalls. The average precision is computed as the mean precision value at 11 equally spaced recall levels (0, 0.1, 0.2, . . . , 1). On PASCAL VOC2007 evaluation measure, mAP is 0.5 when the IoU between prediction bounding box and ground truth box is 0.5. As a result mAP has global view of precision-recall curve. For every recall, the maximum precision is taken. In COCO, mAP is 101-interpolated point computed over ten different IoU (0.5:0.05:0.95) with step size of 0.05. The final value of mAP is average over the classes. In this work, we evaluate the three detection DNNs (SSD, YOLOv3, and Faster-RCNN) and three segmentation models (ANN, U-Net, DeepLabv3+) on a test set of 58 images. The total count of spikes in the test set is 125. The test images contain not just the mature spike in reproductive cycle of wheat but it also include the example of emergent and partially visible spikes, see Fig. 3. These spikes were distributed over 12 images in testing set and the total count is 18. The goal was to see whether the trained model can detect the high-frequency boundary in those spikes.

Evaluation of spike segmentation models

The performance of segmentation method was quantified by commonly used evaluation measure of boundary F1 score (also known as Dice coefficient) and Intersection-Over-Union (IoU) also known as Jaccard index. Average Dice coefficient, aDC is another metric used for pixel labeling calculated by Equation 4 and then taking the average of both regions. Given the set of class of ground truth spike and background label and predicted binary labels, the IoU metric is defined as the number of pixels common between the ground truth and predicted mask divided by the total number of pixels present across both masks. Mean IoU represent average intersection over union of spike and non-spike region. The evaluation of predicted pixel of object is compared with ground truth computed by Equation 10. The output of the segmentation network is binary pixels (spike pixel = 1 and non-spike pixel = 0).

$$IoU = \frac{TP}{TP + FP + FN} \quad (10)$$

Spike detection and segmentation experiments were run on a Linux operating system with Ryzen 7 3800X using 80GB RAM and RTX 2080Ti (8GB VRAM)

Results

The data set of 292 images was divided into training and test sets in the proportion 80:20 regardless of spike numbers, spatial position and orientation. All images were manually annotated for training and testing of spike detection and segmentation models. Consequently, 234 images with the total number of 600 spikes were used for training of DNN and ANN models. The training set was extended with inclusion of YSYC wheat images as shown in Table 2. The number of augmented images used in pattern detection DNNs is summarized in Table 3.

Spike detection experiments

Detection of spike patterns was performed using SSD, Faster-RCNN and YOLOv3/v4 DNN models trained on a data set of totally 234 images as described above. Table 3 gives a summary of evaluation of all spike detection DNN models on PASCAL VOC ($AP_{0.5}$) and COCO detection metrics ($AP_{0.5:0.95}$).

Spike detection using SSD

The SSD model is trained using stochastic gradient descent (SGD) with initial learning rate of 0.001, momentum of 0.9, weight decay of 0.0005, and batch size of 32. SSD was trained for 22000 iterations, which took 10 hours on the GPU. On that iteration, the loss was minimized on validation data and further training overfit the model. Out of three DNNs, SSD performed with lowest average precision. In this regard our observation confirms previous findings from³³ that SSD does not perform well on small objects such as spikes in our case.

Spike detection using Faster-RCNN

The Faster-RCNN was trained for 6000 iterations with binary cross-entropy and learning rate scheduling strategy of exponential decay. The network was developed with Adam optimizer with momentum of 0.9. In training a batch size is set to 6. Inception v2 was taken as backbone for the main detector. Faster-RCNN training was trained for 6500 iterations for nine hours. Around 6000, the loss and in-training AP was lowered enough so that the training process is stopped. Then the training is performed on the training set of images with 800x600 resolution. The number of false positive is 6 on total number of spikes in the test set (125). The number of true positive spikes is 100 on lower resolution, while on the original image resolution, the number of detected spikes is 119 and false positive is 3. Remarkably, the set of false positive spikes comprises mostly of GSGC test images. The inference time of the Faster-RCNN is 0.25 frame/sec on original image resolution. The test images comprise side viewing spikes only. There was no difference of performance by using dropping learning rate on fixed iteration.

Spike detection using YOLOv3/v4

The third DNN we trained is YOLOv3 that has performed well on VOC 2007, 2012, and MS COCO data set. YOLOv3 was trained using stochastic gradient descent (SGD) algorithm for nine hours with batch size of 64 and subdivision of 16. The input height and width to the network was kept on resolution of 416x416. The learning parameter was 0.001 with decay factor of 0.0005 and momentum of 0.9. The inference on the test image was done with non-maximal suppression to exclude multiple prediction on spikes. In addition, YOLOv4 was trained for 20,000 epochs with polynomial decay learning rate scheduling strategy starting at 0.1 with decay of 0.005; momentum of 0.9 and mini-batch size of 8. All detection models had achieved comparatively good result ($AP_{0.5} > 0.75$) with Faster-RCNN outperforming SSD and YOLOv3/v4 models by 20% and 1.06%, respectively. The mAP of YOLOv3 and YOLOv4 are similar on $AP_{0.5}$, but YOLOv4 has shown a better precision on $AP_{0.50:0.95}$. On low resolution images (800x600), the $AP_{0.5}$ of the DNNs decreased by 3-5% which indicate the DNNs didn't extract feature for high frequency spike region in lower resolution images. The inference time of the YOLOv3 on test image is 2.30 frame/sec which is close to average value of YOLOv4.

Among the above three detecting DNNs, Faster-RCNN and YOLOv3 models showed a significantly better performance with mAP over 0.94 compared to SSD having a modest mAP of 0.78. The best models were selected on the basis of $AP_{0.5}$. Fig. 4(a-d) shows examples of Faster-RCNN and YOLOv3 performance on test images of matured spikes. Such spikes were localized by Faster-RCNN, YOLOv3 and YOLOv4 with $AP_{0.5} = 0.99$. However, not all spikes show the same prominent optical appearance as matured spikes growing on the top of the plant. In addition to such clearly visible 'top spikes', some matured spikes may appear in the middle of the mass of leaves that have a similar color fingerprint. Yet another category of spikes represent emergent and occluded spikes that differ from matured spikes with regard to both effectively visible area and texture. Different optical appearance of such spikes leads to a decreased performance of DNNs with YOLOv4 achieving the higher $AP_{0.5} = 0.805$ followed by Faster-RCNN $AP_{0.5} = 0.800$. Examples of detection of occluded and emergent spikes using Faster-RCNN and YOLO are shown in Fig. 4(e-h). Performance measures of all DNNs including AP, accuracy and average probability for matured spikes appearing on the top of the plant, in the middle of the mass of leaves ('inner spikes') as well

as partially visible occluded/emergent spikes are summarized in Table 4. Fig. 5 shows the cumulative confusion matrix for Faster-RCNN and YOLOv3 detection models.

Spike segmentation experiments

Segmentation of spike images was performed using a shallow ANN and two DNN models (U-Net and DeepLabv3+). Table 5 summarizes the performance of all three spike segmentation models on the test set of spike images.

Spike segmentation using ANN

The training of ANN is performed on manually segmented ground truth images where spike pixels have the intensity value 1 and remaining regions - zero. In the test set of spike images, the shallow ANN showed a satisfactory performance with aDC of 0.76 and Jaccard Index of 0.61.

Spike segmentation using U-Net

U-Net was trained on RTX 2080Ti for 45 epochs on 256x256 frames and in training process was validated by binary cross entropy loss and Dice coefficient on 45 images (0.1 *training set) of validation set. No improvement was observed when Tversky loss was used for training process.³⁴ In the test set of spike images, the U-Net showed reached aDC of 0.9 and Jaccard Index of 0.84. The evaluation measures of U-Net in training process are shown in Fig. 6.

Spike segmentation using DeepLabv3+

255 RGB images in the original image resolution of 2560x2976 were used for training and 43 for model evaluation. The evaluation metrics for in-training performance was mean IoU for the binary class labels whereas net loss across the classes is computed from cross-entropy and weight decay loss. ResNet101 was used as backbone for feature extraction. On test set, DeepLabv3+ has shown the highest aDC of 0.935 and Jaccard Index of 0.922 among three segmentation models. Examples of spike segmentation using two best performing segmentation models, i.e. U-Net and DeepLabv3+, are shown in Fig. 7.

Domain adaptation study

To evaluate the generalizability of our spike detection/segmentation models, two independent image sets were analyzed:

- Barley and rye side view images that were acquired with the optical setup including blue background photo chamber, viewpoint and lighting conditions as used for wheat cultivars. This image set is given by 37 images (10 barley and 27 rye) RGB visible light images containing totally 111 spikes. The longitudinal length of spikes in barley and rye is larger than wheat by few centimeter (based on visual inspection).
- Two bushy Central European wheat cultivars (42 images, 21 from each cultivars) imaged using LemnaTec-Scanalyzer3D (LemnaTec GmbH, Aachen, Germany) at the IPK Gatersleben in side view having on average 3 spikes per plant Fig. 9(a) and top view Fig. 9(b) consisting of 15 spikes in 21 images. A particular challenge of this data set is that the color fingerprint of spikes is very much similar to the remaining plant structures.

Evaluation tests with new barley/rye images

Evaluation tests with these new images have shown that YOLOv4 outperforms Faster-RCNN and YOLOv3 measured with regard to F1 score and $AP_{0.5}$ on test set of barley and rye images. On Barley images, YOLOv4 achieved F1 score of 0.92 and $AP_{0.5}$ of 0.88 followed by YOLOv3 with F1 of 0.91 and $AP_{0.5}$ of 0.85. Furthermore, we evaluated the rye images separately on F1 and $AP_{0.5}$. On rye test images, YOLOv4 also performed the highest with F1 score of 0.99 and $AP_{0.5}$ of 0.904, YOLOv3 ($AP_{0.5} = 0.870$) and Faster-RCNN ($AP_{0.5} = 0.605$). The less accurate prediction of Faster-RCNN on the barley and rye is associated with false multiple spike detection (FP). In this case the better performance of YOLOv4 is associated with non-maximal suppression of multiple bounding boxes on a single spike. The detection results of YOLOv4 and Faster-RCNN are depicted in Fig. 4(i-l). When we further test the detection DNNs on overlapping (partially occluded) spikes in barley/rye test set. In most cases, Faster-RCNN produce multiple prediction or false positive while YOLOv3 and its v4 variant performed well on it, see Fig. 8. When U-Net and DeepLabv3+ were tested on barley and rye images, U-Net attained aDC of 0.31 whereas DeepLabv3+ showed and increase of 39% with aDC of 0.43. On overview of model performance on the barley/rye data set is shown in Table 6.

Evaluation tests with images from another phenotyping facility

In addition to images of different grain plants from the same screening facility, evaluation of spike detection models was performed with images from two bushy Central European wheat cultivars that were acquired from another plant phenotyping platform. These evaluation tests have shown that the F1 score of Faster-RCNN on two cultivars was better (0.415) than YOLOv3/v4 (0.22) on bushy cultivars, see examples in Fig. 9. While barley and rye images such as shown in Fig. 4(i-l) closely resemble wheat images that were used for training of DNNs (Fig. 4(a-h)), wheat images from the IPK Gatersleben exhibit quite different phenotype with multiple spikes emerging within a mass of leaves with the same color fingerprint as spikes, see Fig. 9. For these plants, Faster-RCNN turned out to perform better with $AP_{0.5} = 0.41$ than YOLOv4 and YOLOv3 with $AP_{0.5}$ of 0.24 and 0.23, respectively, however, it could mainly detect spikes on the top of the plant (90%) and mostly failed on emerging spikes surrounded or occluded by leaves, Fig. 9(a). Furthermore, DNNs detection models originally trained on side view images were exemplarily tested on top view images of central European wheat cultivars. Due large difference in illumination, spatial orientation, optical appearance, projection area and the overall shape of spikes in the top view differ from side view images that were used for model training. Consequently, Faster-RCNN attained an $AP_{0.5}$ of 0.20 followed by YOLOv4 (0.14) and YOLOv3 (0.10) for this test set of top wheat images. The results of DNN detection model performance on wheat images from another (IPK) screening facility are summarized in Table 6.

SpikeApp demo tool

Three of totally six neural network models investigated in this study, namely, YOLOv3 for spike detection as well as ANN and U-Net for spike segmentation were integrated into a GUI-based software tool (SpikeApp) which demonstrates not only the performance of these three models, but also calculates more than 70 phenotypic traits of detected spikes regions in terms of color, shape and textural descriptors. Fig. 10 shows the screenshot of the SpikeApp, which can be downloaded along with example images from <http://ag-ba.ipk-gatersleben.de/spikeapp.html>.

Discussion

The current study aimed to quantitatively compare the performance of different neural network models for detection and segmentation of grain spikes in visible light greenhouse images.

Predictive power of trained detection models turned out to be different for distinct spike patterns and their position in the plant. Occluded/emergent spikes and inner spikes appearing in the middle of a mass of leaves represent a more challenging problem to DNN models compared to matured top spikes that were predominantly used in this work for model training. In particular, best performing detection DNNs (YOLOv3/v4 and Faster-RCNN) achieved higher accuracy on matured top spikes, while for the group of inner and occluded/emergent spikes the performance of Faster-RCNN was reduced. To improve the model performance on occluded/emergent spikes, more examples of such partially visible spikes should be included in the training image set. Further data augmentation strategies including perturbation of fore- and background colors, rigid and non-rigid geometrical transformations, and permutations of spike and leaves relative positions (neighborhood) should be applied for enhancement of model robustness and applicability. Remarkably, YOLOv4 has the built-in image augmentation methods of Random Erase, CutMix and MixUp which resulted in improved performance by detection of occluded/emergent spikes. However, separation of multiple overlapping spikes remains an unsolved problem which demands special handling. One of the improvement strategies comprises augmenting the spatial location resulting from the detection and dense labeling of segmentation DNN models. To some extent, our spike detection DNNs trained on images of a particular wheat phenotype were capable to provide reliable results on new images of barley and rye plant that exhibit different spike size and texture. However, in general different cereal crops encompass quite large variation of spike color, textural and geometric features that have to be included into the training set in order to achieve more accurate and robust model performance for a variety of possible spike phenotypes.

For the task of pixel-wise spike segmentation, DNN models such as DeepLabv3+ and U-Net have shown superior performance in comparison with the shallow ANN. The segmentation DNNs demonstrated a high accuracy when tested on new images of the same wheat cultivar. However, they shown a modest performance by application to barley and rye images from the same screening facility, and rather poor performance when applied to considerably different wheat cultivars imaged with another screening facility. A particular challenge for DNN segmentation models seems to represent pixels on the spike boundary as they exhibit particularly large variations in color and neighborhood properties depending on the type of grain crops (e.g., top yielding vs. bushy plant phenotype; spike color, texture, size, shape) and scene illumination. Considerably larger set of different plant phenotypes and optical scenes is required to achieve significantly more accurate and robust results of spike segmentation using DNN models.

Considering speed, accuracy and limitations of each single investigated DNN, a combination of YOLO and DeepLabv3+ appears to be a promising approach to further development of a combined framework for spike detection, segmentation and phenotyping.

References

1. Food and Agriculture Organization of the United Nations. Food Outlook, Biannual Report on Global Food Markets (2019).
2. Alexandratos, Nikos and Bruinsma, Jelle. World Agriculture Towards 2030/2050: The 2012 Revision (2012).
3. Guo, Z. *et al.* Manipulation And Prediction Of Spike Morphology Traits For The Improvement Of Grain Yield In Wheat. *Scientific reports* **8**, 1–10 (2018).
4. Minervini, M. and Scharr, H. and Tsafaris, S.A. Image Analysis: The New Bottleneck in Plant Phenotyping. *IEEE Signal Proc Mag* **32**, 126–131 (2015).
5. Hasan, M. M., Chopin, J. P., Laga, H. & Miklavcic, S. J. Detection And Analysis Of Wheat Spikes Using Convolutional Neural Networks. *Plant Methods* **14**, 100 (2018).
6. Grillo, O., Blangiforti, S. & Venora, G. Wheat Landraces Identification Through Glumes Image Analysis. *Computers and Electronics in Agriculture* **141**, 223–231 (2017).
7. Bi, K., Jiang, P., Li, L., Shi, B. & Wang, C. Non-destructive Measurement Of Wheat Spike Characteristics Based On Morphological Image Processing. *Transactions of the Chinese Society of Agricultural Engineering* **26**, 212–216 (2010).
8. Misra, T. *et al.* Spikesegnet-a Deep Learning Approach Utilizing Encoder-decoder Network With Hourglass For Spike Segmentation And Counting In Wheat Plant From Visual Imaging. *Plant methods* **16**, 1–20 (2020).
9. Alharbi, N., Zhou, J. & Wang, W. Automatic Counting Of Wheat Spikes From Wheat Growth Images (2018).
10. Pound, M. P., Atkinson, J. A., Wells, D. M., Pridmore, T. P. & French, A. P. Deep Learning For Multi-task Plant Phenotyping.
11. Qiongyan, L., Cai, J., Berger, B., Okamoto, M. & Miklavcic, S. J. Detecting Spikes Of Wheat Plants Using Neural Networks With Laws Texture Energy. *Plant Methods* **13**, 83 (2017).
12. Narisetti, N., Neumann, K., Röder, M. S. & Gladilin, E. Automated Spike Detection In Diverse European Wheat Plants Using Textural Features And The Frangi Filter In 2D Greenhouse Images. *Frontiers in Plant Science* **11** (2020).
13. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet Classification With Deep Convolutional Neural Networks. In *Advances in neural information processing systems*, 1097–1105 (2012).
14. Everingham, M. & Williams, C. K. The Pascal Visual Object Classes Challenge 2007 (Voc2007) Results .
15. Deng, J. *et al.* Imagenet: A Large-scale Hierarchical Image Database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255 (Ieee, 2009).
16. Tzutalin. Labelimg. Free Software: MIT License (2015). URL <https://github.com/tzutalin/labelImg>.
17. The GIMP Development Team. Gimp (2019). URL <https://www.gimp.org>.
18. Liu, W. *et al.* Ssd: Single Shot Multibox Detector. In *European conference on computer vision*, 21–37 (Springer, 2016).
19. Uijlings, J. R., Van De Sande, K. E., Gevers, T. & Smeulders, A. W. Selective Search For Object Recognition. *International journal of computer vision* **104**, 154–171 (2013).
20. Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. The Pascal Visual Object Classes (Voc) Challenge. *International journal of computer vision* **88**, 303–338 (2010).
21. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-cnn: Towards Real-time Object Detection With Region Proposal Networks. In *Advances in neural information processing systems*, 91–99 (2015).
22. Redmon, J. & Farhadi, A. Yolov3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767* (2018).
23. Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934* (2020).
24. He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
25. Chang, T. & Kuo, C.-C. Texture Analysis And Classification With Tree-structured Wavelet Transform. *IEEE Transactions on image processing* **2**, 429–441 (1993).
26. Christodoulou, C. I., Pattichis, C. S., Pantziaris, M. & Nicolaides, A. Texture-based Classification Of Atherosclerotic Carotid Plaques. *IEEE transactions on medical imaging* **22**, 902–912 (2003).
27. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional Networks For Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241 (Springer, 2015).

28. Kingma, D. P. & Ba, J. Adam: A Method For Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).
29. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. Encoder-decoder With Atrous Separable Convolution For Semantic Image Segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818 (2018).
30. Holschneider, M., Kronland-Martinet, R., Morlet, J. & Tchamitchian, P. A Real-time Algorithm For Signal Analysis With The Help Of The Wavelet Transform. In *Wavelets*, 286–297 (Springer, 1990).
31. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. Semantic Image Segmentation With Deep Convolutional Nets And Fully Connected CRFs. *arXiv preprint arXiv:1412.7062* (2014).
32. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV* (2018). URL <https://github.com/tensorflow/models/tree/master/research/deeplab>.
33. Huang, J. *et al.* Speed/Accuracy Trade-offs For Modern Convolutional Object Detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7310–7311 (2017).
34. Salehi, S. S. M., Erdogmus, D. & Gholipour, A. Tversky loss Function for Image Segmentation using 3D Fully Convolutional Deep Networks. In *International workshop on machine learning in medical imaging*, 379–387 (Springer, 2017).

Author contributions statement

S.U. conceived and performed theoretical and experimental investigations, developed software for data processing, analyzed the data, prepared figures and tables, wrote and reviewed the paper. M.H. developed software for data processing and visualization, performed experimental investigations, reviewed drafts of the paper. N.N. developed software for data processing and visualization, performed experimental investigations, reviewed drafts of the paper. J.H. co-conceptualized the study, reviewed drafts of the paper. E.G. co-conceptualized the study, conceived theoretical and experimental investigations, prepared figures and tables, wrote and reviewed the paper. All authors reviewed the manuscript.

Acknowledgement

This work was supported from European Regional Development Fund-Project **SINGING PLANT(No. CZ.02.1.01/0.0/0.0/16_026/0008446)** which received a financial contribution from the Ministry of Education, Youths and Sports of the CR in the form of special support through the National Program for Sustainability II funds. This work was also supported by the project of specific research provided by the Masaryk University.

Conflict of interest statement

The authors declare no competing interests.

Data availability statement

In addition to data presented in the main text, demo software as well as examples of spike images are provided for download from <https://ag-ba.ipk-gatersleben.de/spikeapp.html>.

Table 1. Overview of 19 Central European wheat cultivars used in this study for training and evaluation of detection and segmentation methods. Other nine cultivars corresponding to the genotype 11 include Pannonia NS, Tobak, Stephens, Cubus Bohemia, Midas, Genius, Fakir, Turandot, Frisky.

Genotypes	Cultivar	Images
1	Avenue	45
2	Elly	14
3	IS Spirella	32
4	Amerigo	13
5	Manager	23
6	Pobeda	24
7	Jindra	23
8	Izvor	29
9	Timber	12
10	Ilona	22
11	Others*	55
Total		292

Table 2. Summary of image sets used for training and testing of detection and segmentation models. the training set for segmentation incorporate additional augmented data.

Total Images	Training set			test set	viewing angle	resolution
	GSGC	YSYC	Non-spike Images			
292	203	27	4	58	side	2560x2976

Table 3. Summary of evaluation of spike detection DNN models on PASCAL VOC ($AP_{0.5}$) and COCO detection metrics ($AP_{0.5:0.95}$), respectively. Best results are highlighted in bold.

Detection DNNs	Backbone	Training set/augmentation	$AP_{0.5}$	$AP_{0.75}$	$AP_{0.5:0.95}$
SSD	Inception resnet v2	234/55	0.780	0.551	0.470
YOLOv3	Darknet53	234/55	0.941	0.680	0.604
YOLOv4	CSPDarknet53	234/ 51	0.941	0.700	0.610
Faster-RCNN	Inception v2	234/55	0.950	0.822	0.660

Table 4. Summary of detection model evaluation on matured top/inner vs. occluded/emergent spikes. Inner spikes include also occluded spikes. Probability, Pr is shown in first column. $AP_{0.5}$ and $AP_r:AP_{0.5:0.95}$ are PASCAL VOC and COCO evaluation measure respectively. The number of top, occluded and inner spikes is 80, 27 and 45. Best results are highlighted in bold.

Methods	Top spikes: 80			Occluded/emergent: 27			Inner spikes: 45		
	Pr	$AP_{0.5}$	AP_r	Pr	$AP_{0.5}$	AP_r	Pr	$AP_{0.5}$	AP_r
SSD	0.81	0.910	0.620	0.59	0.650	0.301	0.70	0.650	0.320
YOLOv3	0.97	0.999	0.708	0.74	0.750	0.450	0.96	0.890	0.500
YOLOv4	0.99	0.999	0.700	0.74	0.805	0.550	0.96	0.880	0.520
Faster-RCNN	0.99	0.999	0.750	0.79	0.800	0.532	0.96	0.910	0.570

Table 5. Summary of evaluation of spike segmentation models. aDC score evaluates the predicted pixel labels with binary ground truth as defined in Section . U-Net and DeepLabv3+ training set include 150 and 43 augmented image on baseline data set of 234 images while no augmentation was used in training ANN. Best results are shown in bold.

Segmentation model	Backbone	Training set/Augmentation	aDC/m.F1	Jaccard Index
ANN	–	234/55	0.760	0.610
U-Net	VGG-16	384/150	0.906	0.840
DeepLabv3+	ResNet101	298/43	0.935	0.922

Table 6. Summary of detection/segmentation of barley/rye spikes using detection/segmentation DNNs. YOLOv3 performed better than Faster-RCNN whereas DeepLabv3+ showed higher aDC having more accurate spike boundaries. The F1 score and $AP_{0.5}$ of new wheat cultivars (top and side view RGB images) from IPK data set is shown.

Methods	Backbone	Barley/rye dataset					bushy wheat cultivars			
		aDC	$F1_{\text{barley}}$	$AP_{0.5:\text{barley}}$	$F1_{\text{rye}}$	$AP_{0.5:\text{rye}}$	$F1_{\text{top}}$	AP_{top}	$F1_{\text{side}}$	AP_{side}
YOLOv3	Darknet53	-	0.91	0.850	0.99	0.870	0.15	0.100	0.25	0.233
YOLOv4	CSPDarknet53	-	0.92	0.880	0.99	0.904	0.20	0.140	0.30	0.240
Faster-RCNN	Inception v2	-	0.80	0.690	0.79	0.650	0.28	0.205	0.55	0.410
U-Net	VGG-16	0.310	-	-	-	-	-	-	-	-
DeepLabv3+	ResNet101	0.430	-	-	-	-	-	-	-	-

Figure 1. Example of spike detection and segmentation in greenhouse wheat images: (a) RGB visible light image of a matured wheat plant, (b) detection of spikes by rectangular bounding boxes, (c) pixel-wise segmentation of spikes.

Figure 2. Comparison of performance of Faster-RCNN vs. YOLOv3: (a) Faster-RCNN in-training loss and average precision (AP) versus Iterations of Faster-RCNN. At 6000 iteration the binary cross entropy loss was minimized with high AP and further training increase loss and AP altogether. (b) YOLOv3 in-training binary cross entropy loss and average precision versus the epoch number.

Figure 3. Examples of spike ROIs from the test image set: (a-d) emergent, partially visible spikes vs. (e,f) matured spikes (GSGC, YSYC).

Figure 4. Detection in grain spikes using pre-trained Faster-RCNN (green bounding boxes) and YOLO (pink bounding boxes) DNN: (a-d) examples of detection of top wheat spikes, (e-h) examples detection of emergent wheat spikes, (i-l) examples of detection barley and rye spikes. White bounding boxes indicate spikes that were not detected by the DNN classifier in this particular image.

Figure 5. Confusion matrix corresponding to IoU=0.75 for (a) Faster-RCNN and(b) YOLOv3.

Figure 6. Comparison of performance of U-Net and DeepLabv3+: (a) Dice coefficient (red line) and binary cross-entropy (green line) over training epochs of U-Net. The training is also validated by Dice coefficient (light seagreen line) and loss (purple line). (b) Training of DeepLabv3+ depicted as function of mean IoU and net loss vs. the number of training epochs.

Figure 7. Examples of U-Net and DeepLabv3+ segmentation of spike images: (a) original test images, (b) ground truth binary segmentation of original images, segmentation results predicted by (c) U-Net and (d) DeepLabv3+, respectively. The predominant inaccuracies in both the NN models are associated with boundary pixels of spike followed false positive.

Figure 8. Examples of detection of overlapping/occluding spikes: (a-c) test results of overlapping spikes in barley/rye using Faster-RCNN. In most cases, Faster-RCNN failed to detect overlapping spikes as separate objects. (d-f) YOLOv3 results on barley/rye data set, in (d) two close spikes were separated.

Figure 9. Examples of application of Faster-RCNN trained on data set in Table 1 for detection of spikes of Central European wheat cultivars in images with different (white) background: (a) DNN failed to detect some spikes in the side view image, (b) early emergent spikes and some matured spike in the top view remained undetected.

Figure 10. Screenshot of the SpikeApp software tool for demonstration of DNN/ANN performance on detection, segmentation and phenotyping of grain spikes. On the left hand side of the tool, the control and parameter section can be found while on the right the output area located. On the right, below of the images, a table with the extracted features for each images is provided to the user to get a quick feedback.

Figures

Figure 1

Example of spike detection and segmentation in greenhouse wheat images: (a) RGB visible light image of a matured wheat plant, (b) detection of spikes by rectangular bounding boxes, (c) pixel-wise segmentation of spikes.

Figure 2

Comparison of performance of Faster-RCNN vs. YOLOv3: (a) Faster-RCNN in-training loss and average precision (AP) versus Iterations of Faster-RCNN. At 6000 iteration the binary cross entropy loss was

minimized with high AP and further training increase loss and AP altogether. (b) YOLOv3 in-training binary cross entropy loss and average precision versus the epoch number.

Figure 3

Examples of spike ROIs from the test image set: (a-d) emergent, partially visible spikes vs. (e,f) matured spikes (GSGC, YSYC).

Figure 4

Detection in grain spikes using pre-trained Faster-RCNN (green bounding boxes) and YOLO (pink bounding boxes) DNN: (a-d) examples of detection of top wheat spikes, (e-h) examples detection of emergent wheat spikes, (i-l) examples of detection barley and rye spikes. White bounding boxes indicate spikes that were not detected by the DNN classifier in this particular image.

Figure 5

Confusion matrix corresponding to IoU=0.75 for (a) Faster-RCNN and (b) YOLOv3.

Figure 6

Comparison of performance of U-Net and DeepLabv3+: (a) Dice coefficient (red line) and binary cross-entropy (green line) over training epochs of U-Net. The training is also validated by Dice coefficient (light seagreen line) and loss (purple line). (b) Training of DeepLabv3+ depicted as function of mean IoU and net loss vs. the number of training epochs.

Figure 7

Examples of U-Net and DeepLabv3+ segmentation of spike images: (a) original test images, (b) ground truth binary segmentation of original images, segmentation results predicted by (c) U-Net and (d) DeepLabv3+, respectively. The predominant inaccuracies in both the NN models are associated with boundary pixels of spike followed false positive.

Figure 8

Examples of detection of overlapping/occluding spikes: (a-c) test results of overlapping spikes in barley/rye using Faster-RCNN. In most cases, Faster-RCNN failed to detect overlapping spikes as separate objects. (d-f) YOLOv3 results on barley/rye data set, in (d) two close spikes were separated.

Figure 9

Examples of application of Faster-RCNN trained on data set in Table 1 for detection of spikes of Central European wheat cultivars in images with different (white) background: (a) DNN failed to detect some spikes in the side view image, (b) early emergent spikes and some matured spike in the top view remained undetected.

Figure 10

Screenshot of the SpikeApp software tool for demonstration of DNN/ANN performance on detection, segmentation and phenotyping of grain spikes. On the left hand side of the tool, the control and parameter section can be found while on the right the output area located. On the right, below of the images, a table with the extracted features for each images is provided to the user to get a quick feedback.