SpikeletFCN: Counting Spikelets from Infield Wheat Crop Images Using Fully Convolutional Networks

Tahani Alkhudaydi[1,2](✉), Ji Zhou[1,3,4], and Beatriz De La lglesia[1]

[1] University of East Anglia, Norwich Research Park, Norwich NR4 7TJ, UK
{t.alkhudaydi,b.iglesia}@uea.ac.uk

[2] Faculty of Computers and IT, University of Tabuk, Tabuk 71491, Saudi Arabia
talkhudaydi@ut.edu.sa

[3] Earlham Institute, Norwich Research Park, Norwich NR4 7UZ, UK
Ji.Zhou@earlham.ac.uk

[4] Plant Phenomics Research Center, China-UK Plant Phenomics Research Centre, Nanjing Agricultural University, Nanjing 210095, China
Ji.Zhou@njau.edu.cn

Abstract. Currently, crop management through automatic monitoring is growing momentum, but presents various challenges. One key challenge is to quantify yield traits from images captured automatically. Wheat is one of the three major crops in the world with a total demand expected to exceed 850 million tons by 2050. In this paper we attempt estimation of wheat spikelets from high-definition RGB infield images using a fully convolutional model. We propose also the use of transfer learning and segmentation to improve the model. We report cross validated Mean Absolute Error (MAE) and Mean Square Error (MSE) of 53.0, 71.2 respectively on 15 real field images. We produce visualisations which show the good fit of our model to the task. We also concluded that both transfer learning and segmentation lead to a very positive impact for CNN-based models, reducing error by up to 89%, when extracting key traits such as wheat spikelet counts.

Keywords: Wheat · Spikelet counting · Plant phenotyping · Image analysis · CNN · Density estimation

1 Introduction

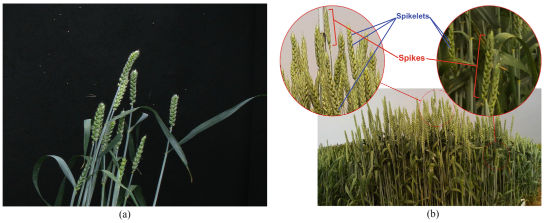
The application of the internet of things (IoT) in agriculture has enabled the monitoring of crop growth through networked remote sensors and non-invasive imaging devices [7,27]. Analysis of the output of such systems with machine learning and image processing techniques can help to extract meaningful information to assist crop management. For example, yield quantification can be tied to other features measured (e.g. temperature, humidity, variety of seed, etc.) to ultimately develop fully automated monitoring systems capable of delivering real-time information to farmers.

© Springer Nature Switzerland AG 2019
L. Rutkowski et al. (Eds.): ICAISC 2019, LNAI 11508, pp. 3–13, 2019.
https://doi.org/10.1007/978-3-030-20912-4_1

Wheat is one of the three major crops in the world with a total demand expected to exceed 850 million tons by 2050 [1]. One of the key challenges for wheat is to stabilise the yield and quality in wheat production [22]. However, climate change and related environmental issues have affected yield production [11].

In this paper, we focus on the task of counting spikelets in wheat images as a form of yield quantification for wheat crops. In particular, we use a density estimation method which has been applied in the context of crowd counting [14], to count spikelets.

The tasks of counting wheat spikelets from infield images (Fig. 1) (as opposed to images obtained in some constrained lab environment) presents some real challenges because of their self-similarity, high volume per image, and severe occlusion as well as the challenges posed by lighting and other variations in the images captured. Image processing or machine-learning approaches for object counting require manual identification of features. Deep learning can automatically extract useful features, and can also lead to high accuracy in image classification tasks [13]. Convolutional Neural Networks (CNNs), a particular type of deep learning model, learn their own features representations and have shown real promise in many areas in computer vision and plant phenotyping [24]. For that reason and because density estimation is considered as structural problem (requiring a prediction for each pixel in the image), we employ a Fully Convolutional Network (FCN) [15] to solve the task.



(a) (b)

Fig. 1. Example of images from (a) ACID dataset and (b) CropQuant dataset which shows spikes, and spikelets.

Furthermore, because the data annotation required to extract 'ground truth' from images is expensive in term of time and resources, we utilised transfer learning in the task of density estimation [18,25]. Transfer learning enhances the image training set with further labelled images from other context and those can be used to pre-train some of the parameters improving the model fit.

Our overall approach is as follows. We employ a fully convolutional model (SpikeletFCN) to perform density estimation from dot annotated images. We utilise additional labelled data for the density estimation by means of transfer learning. In addition, we investigate training SpikeletFCN with and without prior segmentation and compare the performance of each. Section 2 presents research that is relevant to our method. Section 3 discusses the datasets we used, the architecture of SpikeletFCN, model optimisation and training procedure details. Section 4 describes the performance results of testing SpikeletFCN and the their interpretation. Finally, Sect. 5 presents our conclusions.

2 Related Work

Object counting from images is a difficult problem that emerges in many different scenarios, for example, monitoring crowds [26], performing wildlife census [2], counting blood cells in images [8] and others. Supervised counting methods required labelled images with ground truth. Methods for supervised counting include counting by detection or by segmentation, regression based methods such as global regression [3,6,12], local regression [4] and density estimation [14].

Many works [2,19,21] have used detection or segmentation in various ways, but they may require intensive labelling. However, when the only task required is to determine the total number of a certain object in an image rather than detecting them or their position, then counting by regression can be more natural and suitable, specially when the number of objects per image is high. It can be divided into three sub-methods: global regression, density estimation and local regression.

Global Regression often maps global image features to a real number [3,6,12]. However, as stated by Lempitsky and Zisserman [14] extracting these features globally discards information about the location of the objects which may be important in some contexts. Also, sufficient labelled images would be required to represent different counts for training purposes.

Learning to count objects through density estimation regression [14] takes into account the spatial information of objects. Density estimation regression learns mapping from local features into pixel level densities. This gives the advantage of integral density estimation over any image regions. Lempitsky and Zisserman [14] used dot annotations to infer density maps and utilised them as training ground truths by applying a normalised 2D Gaussian kernel. Then, they designed a counting cost function that minimises the distance between the target density map and the inferred ground truth one. Subsequently, Fiaschi et al. [5] used random forest regression, which optimised the training process to predict the density map.

On the other hand, Local Regression [4] predicts the local count of a small region in the image directly without the need to predict a density map. However, it uses the density map in the training stage to infer object counts. Also, it employs the concept of redundant counting to ensure maximum counting precision.

Although it captures the local features of objects, it can be expensive and inefficient in term of time and computational resources.

2.1 Counting in Plant Phenotyping

Counting organs or constituent parts of plants is an essential and important task to be tackled in plant phenotyping. For example, TasselNet [16] was developed to count maize tassels from infield maize crop. TasselNet performs counting by local regression using a deep convolutional neural network-based approach. Pound et al. [18] developed a multi-task deep learning model to count and localise wheat spikes and spikelets, achieving good accuracy. They tested the model on wheat crop images captured in a controlled environment inside a glasshouse. Their problem is therefore similar to ours but simpler given the reduced variation in the controlled laboratory environment as opposed to a real field image. Figure 1 shows both type of images.

Also, Madec et al. [17] investigated counting spikes from infield wheat crop images captured by UAV platform using two CNN-based models. The first was Faster-RCNN [20], a CNN based object detection model. The second was an adaptation of TasselNet [16] for this task. They concluded that both models achieved similar results when tested on images containing crops that have a similar distribution of spikes as the images both models trained on. However, they found that Faster-RCNN outperformed other models when tested on images containing more mature crops.

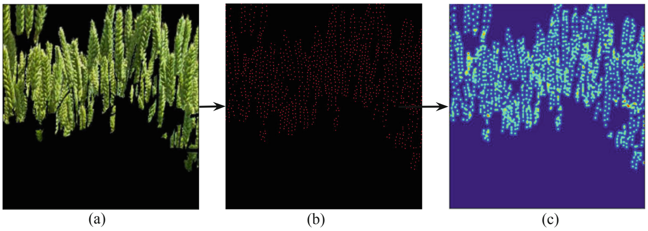


Fig. 2. An example of spikelets density generation where: (a) represents sub-image of a wheat crop, (b) represents corresponding dot annotation and (c) the generated density map from the dot annotation.

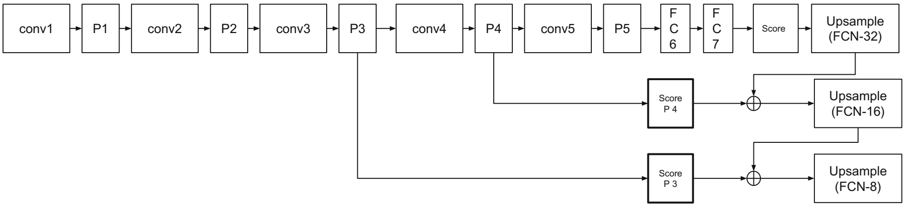


Fig. 3. SpikeletFCN architecture

Also, Hasan et al. [9] tackled the problem of counting spikes by using an R-CNN object detector. They trained four versions of R-CNN on four different growth stages of infield wheat images that vary in growth stage and variety and reported good results.

3 Spikelets Counting Using SpikeletFCN

3.1 Problem Statement

We propose to model the problem of counting spikelets as a density estimation problem. Given N input images $I_1, I_2, \ldots, I_N$ with a size of $H \times W \times D$ that represent infield wheat crop plots, for each image, I_i, there is a corresponding dot map P_i that can be represented as a set of 2D points $P_i = \{P_1, \ldots, P_{SPC_i}\}$, where $|P_i|$ is the number of spikelets in image I_i. Each point is placed at the centre of each spikelet as shown in Fig. 2(b). To generate the ground truth map GT_i (shown in Fig. 2(c)), a 2D Gaussian kernel $\mathcal{N}(p; P, \sigma^2 1_{2\times2})$ is applied to the dot map P_i which generates a density for each pixel p of image I_i. Therefore, the size of GT_i is the same as the input image: $GT_i = \{D^{P_1}, \ldots, D^{P_{H\times W}}\}$ where D^{P_j} is the generated density for the j^{th} pixel in image I_i.

The effect of applying the Gaussian kernel is that it can reflect the crowding around a spikelet by taking into account the information of the pixel's neighbourhood when updating its density value. In other words, the more spikelet occlusion in a certain region, the high density values will be assigned to pixels in the region.

The total number of spikelets in a certain image I_i is the sum of all pixels densities in GT_i:

$$|P_i| = SPC_i = \sum_{p \in I_i} D^p \tag{1}$$

3.2 Datasets

CropQuant Dataset [27]. We used 15 high-dimensional RGB image series of 6×1.5 m wheat plots collected at Norwich Research Park (NRP) between May and July 2016. The image series covers one growing stage: flowering. The resolution of images is 2592 by 1944 pixels, which were captured hourly by R-pi camera modules integrated in the CropQuant workstation. Image data were synchronised with HPC data storage infrastructure at NRP. We have dot annotated each image by placing a dot in the centre of each spikelet. The total number of spikelets in all images is 63,006 and the average spikelet number per scene is 4200.4 with a standard deviation of 197.

ACID Dataset. The Annotated Crop Image Dataset (ACID) has 520 images of wheat plants captured from 21 pots in a glasshouse with a resolution of 1956×1530. The imaging is done by 12 MP cameras and all images have a black background. The images show different spike arrangements and leaves and were obtained in consistent lighting. Also, the images were dot annotated by placing a dot in the centre of each spikelet. The total number of spikelets in all images is 48,000 and the average spikelet number per scene is 92.3 with a standard deviation of 28.52.

Figure 1 shows examples of both CropQuant and ACID images which exemplify their similarities and differences.

3.3 SpikeletFCN Architecture

In our approach, we apply a fully convolutional network to tackle the problem of spikelet counting. Figure 3 represents our architecture. The last fully connected layers attached in any CNN-based classifiers are converted to convolutions. This ensures that the semantics of target objects are preserved which are essential for tasks that require structural predictions (predictions for each pixel) because converting those layers to convolutions provide localisation and shape information about target objects. Our model, SpikeletFCN, is composed of a Very Deep Convolutional Network (VGG16) [23] (Fig. 3: conv1-P5), formed by two fully convolutional (Fig. 3: FC6 and FC7) layers and three upsampling layers. The filter size selected for all convolutional layers is 3 × 3 with a stride of 1 and the max-pool layers have a pooling size of 2 × 2 with a stride of 2. We employ the concept of feature fusion by adding two skip connections (Fig. 3: after P3 and P4) to fuse the local features related to spikelets from lower layers to other shape and semantic features related to the wheat crops from higher layers. We added upsampling layers to ensure we recover the original image size affected by the application of repetitive convolutions and subsampling which reduces the input size.

We found that using a pixel-wise $L2$ loss function (Eq. 2) as the cost function for model optimisation gave the best results to regress the per pixel density:

$$\mathcal{L} = \sum_{p \in I_i} (D^p_{GT_i} - D^p_{predicted})^2 \tag{2}$$

where $D^p_{GT_i}$ is the density ground truth and $D^p_{predicted}$ is the predicted density for a certain pixel p in image I_i.

The weights were updated for every learning iteration using a mini-batch RMSprop optimising algorithm [10] with a learning rate of 0.001 and mini-batch of 20.

3.4 Experimental Set Up

We first formed the training and validation set from the ACID dataset according to the 80:20 split rule. Then, we randomly sampled sub-images with a size of 512 × 512 for each set. After that, we manually selected 1241 sub-images from the training set and 303 sub-images from the validation set that contain spike regions. With those images we trained the model for 100 epochs for the transfer learning experiments described below.

On the other hand, for the CQ_2016 dataset, the limitations imposed by the task of dot annotating, which is time consuming and could therefore only be accomplish for a very reduced number of images, meant we only had 15 images dot annotated for our experiments. We therefore decided to divide them into 3-folds for cross validation, with 5 images per fold. Then, we randomly subsampled 512 × 512 sub-images from each fold individually.

To investigate whether segmenting spike regions could enhance the spikelets counting task, we manually remove the background using ground truth masks. In future research, we intend to also use a CNN to tackle the segmentation, instead of a manual approach.

We trained the model on each fold of the CQ_2016 dataset, validated and tested on the other two folds in four steps:

1. We first trained the model from scratch on the original images (no segmentation) and the model converged after an average of 155 epochs.
2. We then trained the model from scratch on the images with the spike regions isolated so after this manual segmentation the model converged after an average of 75 epochs.
3. We then loaded parameters learned from training the model on the ACID dataset, as described earlier, and continued fine tuning the model using the original images. The model converged after an average of 36 epochs. This represents transfer learning, using the ACID dataset in the initial stage of parameter initialisation and the CQ_2016 dataset to train the final model.
4. We repeated the previous transfer learning model building step, but then combined it with continued fine tuning on the CQ_2016 dataset images with the spike regions isolated and the model converged after an average of 30 epochs.

For the testing phase, SpikeletFCN predicts the density of each pixel in a certain image. Then, the number of spikelets in the image is calculated by summing all the predicted densities over the whole image according to Eq. 1 in Sect. 3.1.

4 Results

Object counting methods use two evaluation metrics to measure the model performance when applied on testing images: mean absolute error (MAE) and means square error (MSE).

We have calculated the cross-validated performance of the SpikeletFCN model for the different experimental steps described in Sect. 3.4 using the MAE and MSE measures. Table 1 shows our results. Table 1 shows that applying segmentation before counting has decreased the spikelet counting error to 82.2 and 102.0 for MAE and MSE respectively when training SpikeletFCN from scratch. This represents a reduction of 83.5% and 81.2% respectively for MAE and MSE with respect to error measures without segmentation.

In terms of transfer learning, loading ACID pre-trained parameters has a positive impact on the model performance by decreasing MAE and MSE to 53.0 and

Table 1. The MAE and MSE of estimating the number of spikelets for two experimental setups (as columns): training SpikeletFCN from scratch and by loading ACID dataset learned parameters and for pre-segmenting images (in rows) on CQ_2016 images.

	Scratch		ACID [18]	
	MAE	MSE	MAE	MSE
With segmentation	82.2	102.0	53.0	71.2
Without segmentation	498.0	543.5	77.12	107.1

71.2 respectively when segmentation is also applied. This represents a decrease of 35.5% and 30.2% respectively when segmentation is applied with respect to the error from the scratch model. When no segmentation is applied, the transfer learning reduces error by 84.5% and 80.3% respectively for MAE and MSE. Hence both segmentation and transfer learning have a very significant effect on error rates. It is worth noting that the pre-trained ACID model has minimised the gap between the SpikeletFCN performance with and without segmentation. The difference in missed spikelets when training from scratch is 415.8 for MAE and 441.5 for MSE. On the other hand, the comparative difference when loading pre-trained ACID parameters is 24.12 for MAE and 35.9 for MSE. Overall, the difference between the best model (with segmentation and transfer learning) and the worse (the scratch model without segmentation) is over 89% for MAE and over 86% for MSE.

In term of model training time, we can infer from Sect. 3 that loading ACID pre-trained parameters and training the model in images with segmentation have resulted on faster training of the model.

Also, we analysed the results in more detail through visualisation. Figure 4 shows some images with their density maps and respective spikelet counts. They show that visually the density maps obtained appear to be reasonably accurate with respect to the original images and seem to improve with the segmentation, though in some cases the prediction represents under or over-counting.

More detailed visual analysis, in this case for the ACID images, is shown in Fig. 5. By comparing the density maps generated from our models (column (b) of Fig. 5 with the 'ground truth' density maps derived from the dot annotation (column (c)) we can note that in some images, SpikeletFCN may be considered as over-counting because it is able to detect spikelets that were miss-annotated (missed) by accident in the dot annotation. For example, in Fig. 5, SpikeletFCN predicted spikelet number for the second and third images as 53.47 and 49.93 while the ground truth for both images was 46.0 and 46.61. However, in these images, spikes that appear to contain a single row of spikelets in the dot annotation are recognised as having more spikelets by the SpikeletFCN model and this seems to correlate to the images in column (a). We can assume that as the dot annotation gets much more complex in the very crowded infield images, dot annotation may also be more inaccurate, so some of our errors may reflect the inaccuracies of our ground truth.

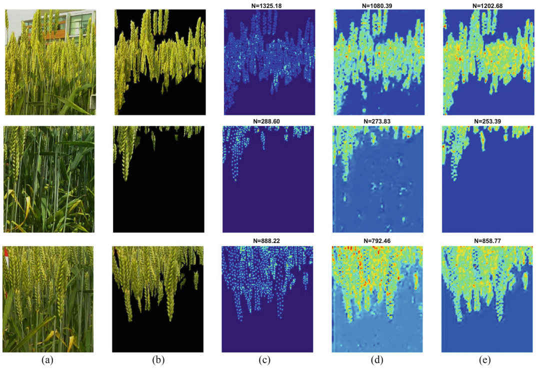


Fig. 4. Visualisation of density map results of testing SpikeletFCN on some CQ_2016 sub-images where (a) represents image patch, (b) image patch without background (c) ground truth for spikelet density map and counts, and (d) and (e) are predicted spikelet density map and count for the original image patch and image patch without background respectively.

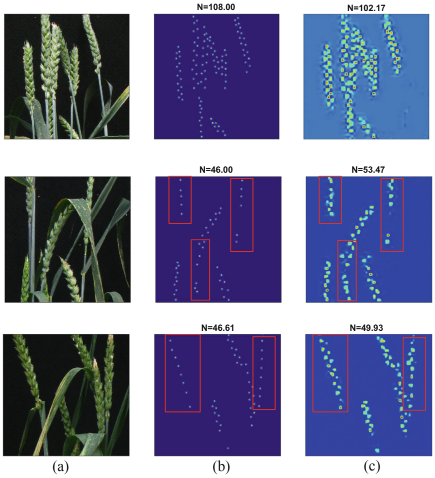


Fig. 5. Visualisation of density maps resulting from testing for the Adapted SpikeletFCN on ACID dataset. (a) is image patch, (b) is 'ground truth' spikelet density map and count obtained after dot annotation, and (c) is the predicted spikelet density map and count.

5 Conclusion

Counting spikelets from infield wheat crop images is a vital step in quantifying yield traits but is very challenging given the variability, density and occlusion associated with spikelets in real wheat images. In this paper, we trained and tested SpikeletFCN to count spikelets using a density estimation approach. We also attempted to improve our learning by applying transfer learning and segmentation.

Our experimental results were very promising and resulted in good error rates, much improved by using both manual segmentation and transfer learning. In particular transfer learning did help to improve the performance of the models trained on infield crops images. Error rates decreased by over 81% when using manual segmentation and over 86% when combining segmentation with transfer learning. Also, it led to faster training of the model.

Visualisation helped us to discover that the process of obtaining ground truth by dot annotation is imperfect and models may actually uncover spikelets which have not been dot annotated. This is encouraging as it means the model is able to learn features of the spikelets, even in the context of imperfect training data.

In the future, we plan to test our model on more infield wheat crops that vary in year growth, growth stages and other factors. We will also develop CNN-based models to tackle the task of spike segmentation as we have shown that it can play an important role in improving the task of spikelets counting.

References

1. Alexandratos, N., Bruinsma, J.: World agriculture towards 2030/2050. Land Use Policy **20**(4), 275 (2012)
2. Arteta, C., Lempitsky, V., Zisserman, A.: Counting in the wild. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 483–498. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_30
3. Cho, S.Y., Chow, T.W., Leung, C.T.: A neural-based crowd estimation by hybrid global learning algorithm. IEEE Trans. Syst. Man Cybern. Part B Cybern. **29**(4), 535–541 (1999)
4. Cohen, J.P., Boucher, G., Glastonbury, C.A., Lo, H.Z., Bengio, Y.: Count-ception: counting by fully convolutional redundant counting. In: 2017 IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 18–26. IEEE (2017)
5. Fiaschi, L., Koethe, U., Nair, R., Hamprecht, F.A.: Learning to count with regression forest and structured labels. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 2685–2688. IEEE (2012)
6. Giuffrida, M.V., Minervini, M., Tsaftaris, S.A.: Learning to count leaves in rosette plants. In: Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP), pp. 7–10 (2016)
7. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. **29**(7), 1645–1660 (2013). https://doi.org/10.1016/j.future.2013.01.010
8. Habibzadeh, M., Krzyżak, A., Fevens, T.: White blood cell differential counts using convolutional neural networks for low resolution images. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013. LNCS (LNAI), vol. 7895, pp. 263–274. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38610-7_25
9. Hasan, M.M., Chopin, J.P., Laga, H., Miklavcic, S.J.: Detection and analysis of wheat spikes using convolutional neural networks. Plant Methods **14**(1), 100 (2018). https://doi.org/10.1186/s13007-018-0366-8

10. Hinton, G., Srivastava, N., Swersky, K.: Neural networks for machine learning. In: Lecture 6a: Overview of Mini-batch Gradient Descent, p. 14 (2012)
11. Howden, S.M., Soussana, J., Tubiello, F.N., Chhetri, N., Dunlop, M., Meinke, H.: Adapting agriculture to climate change. Proc. Nat. Acad. Sci. U.S.A. **104**(50), 19691–19696 (2007). https://doi.org/10.1073/pnas.0701890104
12. Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 3, pp. 1187–1190. IEEE (2006)
13. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
14. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Advances in Neural Information Processing Systems, pp. 1324–1332 (2010)
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation (2015)
16. Lu, H., Cao, Z., Xiao, Y., Zhuang, B., Shen, C.: TasselNet: counting maize tassels in the wild via local counts regression network. Plant Methods **13**(1), 79 (2017)
17. Madec, S., Jin, X., Lu, H., De Solan, B., Liu, S., Duyme, F., Heritier, E., Baret, F.: Ear density estimation from high resolution RGB imagery using deep learning technique. Agric. For. Meteorol. **264**, 225–234 (2019)
18. Pound, M.P., Atkinson, J.A., Wells, D.M., Pridmore, T.P., French, A.P.: Deep learning for multi-task plant phenotyping. In: 2017 IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 2055–2063. IEEE (2017)
19. Ren, M., Zemel, R.S.: End-to-end instance segmentation with recurrent attention. In: CVPR (2017)
20. 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, pp. 91–99 (2015)
21. Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using multiple local features. In: Digital Image Computing: Techniques and Applications, 2009, DICTA 2009, pp. 81–88. IEEE (2009)
22. Shewry, P.R.: Wheat. J. Exp. Bot. **60**(6), 1537–1553 (2009)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR
24. Tardieu, F., Cabrera-Bosquet, L., Pridmore, T., Bennett, M.: Plant phenomics, from sensors to knowledge. Curr. Biol. **27**(15), R770–R783 (2017)
25. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)
26. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 833–841 (2015)
27. Zhou, J., et al.: CropQuant: an automated and scalable field phenotyping platform for crop monitoring and trait measurements to facilitate breeding and digital agriculture. bioRxiv. https://doi.org/10.1101/161547