Noname manuscript No.
(will be inserted by the editor)

Counting Spikelets from Infield Wheat Crop Images Using Fully Convolutional Networks

Tahani Alkhudaydi · Beatriz De La lglesia

Received: date / Accepted: date

Abstract Wheat is one of the world's three main crops, with global consumption projected to reach more than 850 million tons by 2050. Stabilizing yield and quality of wheat cultivation is a major issue. With the use of remote sensing and non-invasive imaging technology, the internet of things (IoT) has allowed us to constantly monitor crop development in agriculture. The output of such technologies may be analyzed using machine-learning algorithms and image processing methods to extract useful information for crop management assistance. Counting wheat spikelets from infield images is considered one of the challenges related to estimating yield traits of wheat crops. For this challenging problem, we propose a density estimation approach related to crowd counting, SpikeCount. Our proposed counting methods are based on deep learning architectures as those have the advantage of being able to identify features automatically. Annotation of images with the ground truth are required for machine learning approaches, but those are expensive in terms of time and resources. We use Transfer Learning in both tasks, segmentation and counting. Our results indicate the segmentation is beneficial as focusing only on the regions of interest improves counting accuracy in most scenarios. In particular, Transfer Learning from similar images produced excellent results for the counting task for most of the stages of wheat development.

Keywords Wheat · Spikelet Counting · Plant Phenotyping · Soft Computing · Image Analysis · CNN · Density Estimation

T. Alkhudaydi
University of Tabuk
Tabuk, 71491, SA
E-mail: talkhudaydi@ut.edu.sa
Corresponding author

B. De La lglesia
University of East Anglia
Norwich Research Park, Norwich, NR4 7TJ, UK
E-mail: b.iglesia@uea.ac.uk

1 Introduction

Object counting from images is a challenging problem that arises in a variety of situations, including crowd surveillance [7, 35, 50], ecological census [4], and counting blood cells in images [16].

Sometimes, the problem is framed in the context of counting objects in still images [3, 4, 8] whereas in other cases, the problem involves counting moving objects in videos [13].

Counting objects can be achieved by a number of approaches including counting by detection [25, 46], segmentation [36] and approaches based on regression, such as global regression [9, 15, 20], local regression [10], and density estimation [23].

The approach we propose in this paper involves the use of a density estimation method, a well-known technique in the context of crowd counting [23], and we apply it to the complex problem of count spikelets in real infield wheat images.

Counting wheat spikelets task from infield images (as opposed to images captured in a controlled laboratory setting) presents some significant hurdles due to their self-similarity, high volume per image, and extreme occlusion, as well as the issues given by lighting and other imaging alterations.

Another challenge is that current methods for counting and related tasks in image analysis often require manual feature identification before applying image processing or machine learning techniques. This relies on expertise of the crop specialists and thus the feature engineering approach may differ from one problem to another.

An alternative powerful approach which we employ to avoid the feature identification problem is to apply deep learning to automatically extract useful features. Deep learning is based on learning data representations through the application of multi-layered neural networks, so features do not need to be identified beforehand. Also, it can lead to a high degree of accuracy in image classification tasks [22].

Convolutional Neural Networks (CNNs), a subset of deep learning models, can autonomously train their own features and generate distinct representations from raw images, and have shown great promise in a variety of areas in computer vision and plant phenotyping [42]. As a result of this, and because density estimation is a structural problem (requiring a prediction for each pixel in the image), we tackle the task using a Fully Convolutional Network (FCN) [26].

Furthermore, as data annotation requires us to extract 'ground truth' from images, particularly in the context of complex counting problems, it is expensive in terms of time and resources. Therefore, we employed Transfer Learning [34, 48] to accomplish the task of density estimation because of a lack of time and resources. This enables us to use already labelled images from other context to help build our model. Transfer learning enhances the training set of images with additional labelled images, which can be utilised to pre-train some of the parameters, thereby improving the model fit and reducing training time.

Therefore, the contribution of this paper is as follows:

– We produce a high-quality high-resolution dataset that consists of three sequences : CQ_2015, CQ_2016, and CQ_2017. It represents three growing seasons of four near isogenic lines (NILs) of bread wheat that were cloned at John Innes Centre (JIC) [39, 40]. They were monitored by distributed CropQuant

workstations [51] in real field environments and measured manually during the key growth stages in wheat growing seasons from 2015 to 2017. Each sequence is composed of 30 images divided into four key growth stages: booting, heading, flowering and grain filling.
- Each sequence is annotated with two types of labels : the spike regions in each image have been labelled and each spikelet in each image was annotated with a dot at the centre. We make the data available through `https://github.com/tanh86/SpikeProject`.
- We develop a deep learning solution (SpikeCount) to tackle the problem of counting spikelets from infield images.
- We also investigate the efficacy of eliminating the background and isolating the region of interests (ROI), which is the spike region in our case.
- We investigate whether the knowledge transferred from similar but yet simpler images captured in a more controlled settings can improve application to spikelet counting.

Note that this extends our previous work [2] where we first introduced and experimented with the counting approach, although in this paper we provide a more robust dataset for validation, an enhanced deep learning solution and new comprehensive results. Furthermore, we provided details of our semantic segmentation deep-learning pipeline in our previous paper [1].

The outline of this paper is as follows: Section 2 presents research that is relevant to the problem and our methods. Section 3 summarises the datasets we used. Section 4 presents the architecture of SpikeCount, details on the model optimisation and training procedures. Section 5 presents the full experimental setups of this work. Section 6 describes the performance results of testing SpikeCount and the interpretation for each experimental setup. Also, we discuss results of testing SpikeCount for each growth stage. Finally, Section 7 presents our conclusions from this experiments.

2 Related Work

2.1 Object Counting Methods

Counting objects can be achieved by a number of approaches, including for example counting by detection of objects [4]. However, detection could be challenging particularly if there are overlapping objects causing occlusion, differences in scale, or a lack of detectable background between distinct objects. Detection of humans is often used in crowd counting [25, 46] but humans can adopt many different postures which may make human body recognition more challenging. Segmentation can be used in various ways to aid to the counting task. For instance, Chan et al. [6] used foreground segmentation to segment the crowd into homogeneous groups and applied regression to estimate the number of people in each group.

When the task is to count the number of instances of a particular object in an image rather than detection, counting by regression may be easier due to the difficulty of the detection problem. Also, it may be the only way if the image suffers from severe object overlapping or if the resolution of the image is low. Counting by regression often maps image global and/or local features to the total number of objects from the whole scene or various extracted parts of the scene. Extracting

these features globally discards information about the location of the objects. Also, an image level labelling is mandatory to represent the total number of objects for training. Thus, it would require a very large number of labelled training samples, which may not be available [23]. Unlike other counting regression methods that use global region features, counting using density estimation regression incorporates the spatial information included in the objects. This would provide some information about the location of the objects in the image. Regression on density estimation learns a mapping from local features to pixel level densities. This enables integral density estimation to be applied to any image region. In other words, it is capable of calculating the density of any location inside an image. By applying a normalised 2D Gaussian kernel to dot annotations, Lempitsky and Zisserman [23] inferred density maps and utilised them as training ground truths. They then developed a cost function for counting that minimises the distance between the desired density map and the inferred ground truth. Subsequently, Fiaschi et al. [12] utilised random forest regression, which optimises the training procedure to predict the density map. Exploiting the advantage of integral density estimation over any sub region of the image, a counting system was presented that estimates object counts interactively [3].

Pham et al. [32] proposed a patch-based method to solve crowd density estimation exploring structured learning using random decision forests. A fast density estimation based visual object counting (DE-VOC) was presented which leveraged the mapping between images and their corresponding density maps in two different features spaces [45]. Xu et al. [47] proposed that using more image features can lead to more robust crowd density estimation models. Also, they utilised more efficient random projection forests to reduce the dimensionality resulting from traditional random forests.

2.1.1 Counting using deep learning algorithms

Convolutional Neural Networks (CNN), which are capable of learning their own features, are beginning to show real promise in the area of object counting [4, 8, 50]. Seguí et al. [38] argue that it is possible to learn to count using CNN focusing on the multiplicity of the object of interest, without annotating the objects via detection or bounding boxes. They achieve this in their CNN by using a two block structure, with the first block made up of two or more convolutional layers which tries to capture and represent the concept they are counting and a second block of fully connected layers that produces the counts.

Arteta et al. [4] used foreground segmentation to aid the learning of density estimation. They trained a segmentation map and a density map jointly to count penguins. French et al. [13] used segmentation to count fish in a two stage process:(1) foreground segmentation and (2) CNN edge detector to accurately segment the severely occluded fish areas.

A recent work by Ren and Zeme [36] has used deep learning algorithms such as Recurrent Neural Nets(RNNs) and CNN to adopt the visual attention that humans use when counting. They locate and segment each object iteratively and keep track of them.

Chattopadhyay et al. [8] explore applying deep learning to count different types of objects in everyday scenes. They solve the counting problem by dividing it into sub-problems on different cells in a non-overlapping grid and then adding the

partial counts. They also take care of context across cells to improve their count. However, they require annotated scenes as input with object bounding boxes and they also require category wise counts for each type of object.

A problem in crowd counting is that small changes to the scene may make the learned models useless for new data as models are often scene specific (e.g. [44, 50]). For example, changes in lighting, resolution, distribution of crowds, and perspective can make models poor in new scenes.There have been a number of attempts to solve the problem of scale variation and perspective distortion [4, 50].

2.1.2 Object Counting in plant phenotyping

Counting the constituent parts of plants is a necessary and critical task in plant phenotyping. For instance, TasselNet [27] was designed to count maize tassels in images of infield maize crops taken between 2010 and 2015 in different places in China. TasselNet performs local regression counting through the use of a deep convolutional neural network-based technique. It was evaluated on eight test sequences and obtained positive outcomes. Also, three public plant counting data sets and a new unmanned aircraft vehicle (UAV)-based data set of maize tassels, wheat ears, and rice plants are used to test TasselNetV3 [28]. To increase model performance, they employed guided upsampling and background suppression.

Another example of counting in plant phenotyping closer to our own application is counting spikes and spikelets in wheat. Spikes and spikelets represent important yield traits for wheat [24]. In this context, Pound et al. [34] developed a multi-task deep learning model for counting and localizing wheat spikes and spikelets with a 95.91 % and 99.66 % accuracy respectively using non-maximal suppression (NMS). They tested the model using images of wheat crops taken in a controlled environment inside a glasshouse. Thus, their task is similar to ours but simpler, due largely to less fluctuation in a controlled laboratory environment versus a real field image.

Additionally, Madec et al. [29] used two CNN-based models to evaluate counting spikes from infield wheat crop images acquired by a UAV platform. The first was Faster-R-CNN [37], an object detection model based on CNNs. The second was a task-specific version of TasselNet [27], a CNN-based local regression model designed for counting maize tassels in infield images. They concluded that both models performed similarly well when tested on images of crops with a comparable distribution of spikes to those used to train them. They discovered, however, that when tested on images having more mature crops, Faster-RCNN outperformed other models.

Fernandez-Gallego et al. [11] developed an image processing pipeline to count and localise wheat spikes (also known as ears) by segmentation. They tested their pipeline on infield durum wheat crops with two growth stages and reported high success rate (over 90%) between algorithm count and manual (image based) counts.

Also, Hasan et al. [17] used an R-CNN object detector to solve the task of counting spikes. They trained four different versions of R-CNN on images of infield wheat at four distinct growth stages. They again observed good findings, with detection accuracy ranging from 88% to 94% across several test image sets.

3 Datasets

3.1 Wheat field experiments

The description of our data was already presented in our previous papers [1, 2] but we include it here for completeness.

To evaluate key yield-related traits in UK bread wheat, we conducted field experiments using four near isogenic lines (NILs) representing genetic and phenotypic variation with a similar genetic background known as"Paragon," an elite UK spring wheat that is also used in the Biotechnology and Biological Sciences Research Council's (BBSRC) Designing Future Wheat (DFW) Programme. The four NILs include the Paragon wildtype (WT), Ppd (photoperiod insensitive), and Reduced Height (Rht) genes (semi-dwarf) genotypes cloned at the John Innes Centre (JIC) [39, 40], which were monitored and measured manually during key growth stages in wheat growing seasons from 2015 to 2017.

3.2 Image acquisition

The high-quality RGB image series used in this work were acquired during a three-year field experiment from 1.5-metre-wide (5-metre-long) wheat plots. Four CQ workstations were dedicated to conducting high-frequency (one image per hour) and high-resolution (2592x1944 pixels) imaging in order to acquire target yield-related trait expression. Each CQ workstation or terminal is comprised of a single-board computer, climate sensors, a sensors integration circuit, an imaging sensor, local storage, network components, a Debian operating system, and a custom-written Python software package for cropping images and collecting climate data[52]. Between May and July, CQ devices generated about 60 GB of image datasets encompassing three growing seasons (i.e. from booting ,GS41–GS49, to grain filling stages ,GS71–GS77). For each growth season, 30 typical images were chosen for phenotypic analysis using deep learning.

To maintain consistent contrast and clarity of wheat images under varying lighting conditions in the field, the latest versions of the open-source picamera imaging library [33] and Scikit-image [43] were used to automate white balance, exposure mode, shutter speed, and calibration adjustments during image acquisition [52]. Using the Internet-of-Things-based CropSight technology, in-field image datasets were synchronized with centrally stored data at Norwich Research Park (NRP). Figure 1 illustrates examples of wheat plot images acquired by CQ workstations from 2015 to 2017 (in columns), indicating that the images selected for yield-related trait analyses were taken under a variety of in-field illumination and weather conditions, and included a variety of background objects throughout the experiments.

3.3 Wheat growth dataset for training, validation, and testing

The images were acquired over three consecutive years and cover four important growth phases (Figure 1). We chose the 2015 sequence to train the models due to the this series having consistent clarity and contrast. The 2016 sequence is then

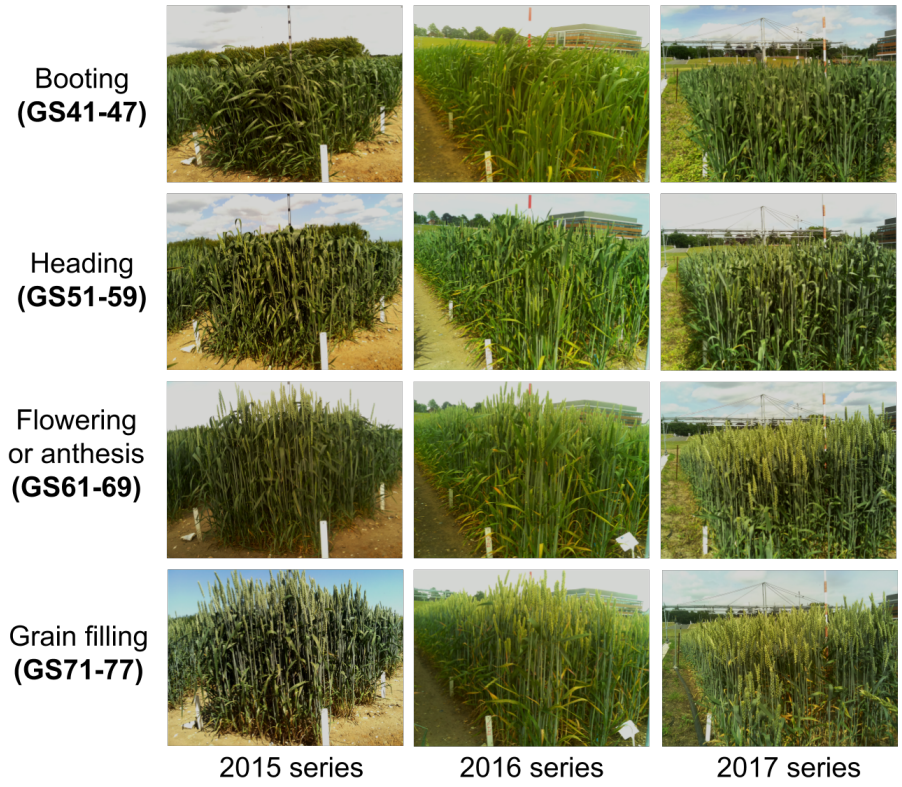


Fig. 1: Wheat growth image series in the field collected by CropQuant workstations, from 2015 growing season to 2017 growing season, covering booting to grain filling growth stages

used to validate our learning model, and the 2017 sequence is used to test the model. This training strategy provides a rather robust validation of our model's performance, as the unseen sequence in 2017 is used to assess the model's generalisation. Figure 2 depicts the distribution of selected images during each growing season (30 images per year, 90 in total). Flowering has the most images (37 out of 90), followed by heading (22 images), grain filling phases (19 images), and booting (12 images). The rationale for this arrangement is that the flowering stage corresponds to the phase during which spikes fully emerge, whereas wheat spikes are typically buried during the booting and heading stages (i.e. GS41-598). It's worth noting that the 2015 sequence lacks booting images because to the transient nature of wheat booting, which typically lasts 1-2 days. As such, it is an intriguing test case for us to train a deep learning model that can segment spike areas gathered over numerous years and at different stages of the ear emerging process (e.g., spikes may only be partially developed) under tough in-field lighting conditions. Figure 3 illustrates a 2017 sequence image with associated labels, spike area masks, and spikelet density maps.

3.4 Wheat growth dataset Labelling

Using the UEA Computer Vision - Image Labelling Tool [14], we have manually labelled images with polygonal labels for the segmentation task and dot annotation for density estimation task. The tool runs as a web-based application and is written in Python 2.7. All images that need to be labelled are stored in a certain folder. Then, the tool detects and load these images. The tool provides a number of options to label the image. We selected the polygonal option to trace the spike shape. Once the tracing is done, we then select the appropriate label and then the corresponding coordinates of polygons are stored in matched encoding file for each image. For the density estimation task, we adjusted the tool to make the dot annotation option - which are represented by a circle- by reducing the size of the circle. The centre coordinates of each circle, representing each spikelet, are stored in the corresponding encoding file for each image.

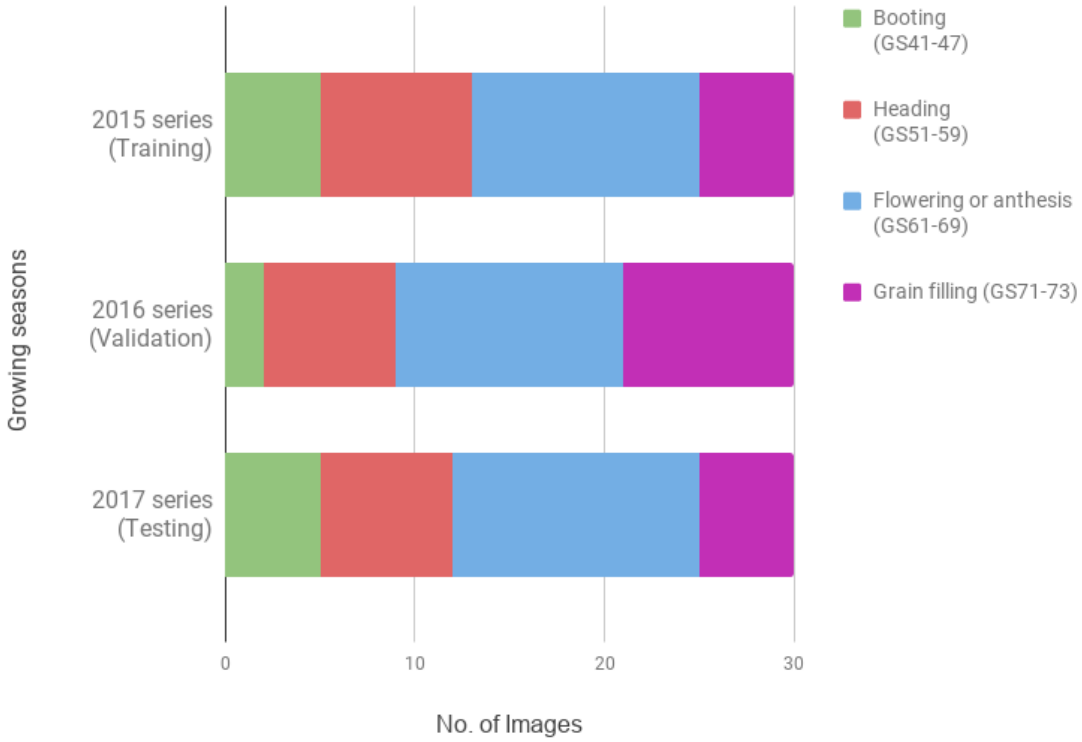


Fig. 2: The distribution of selected images in each growth stage between 2015 and 2017 growing season, which are used for training, validation and testing when establishing the deep-learning architecture

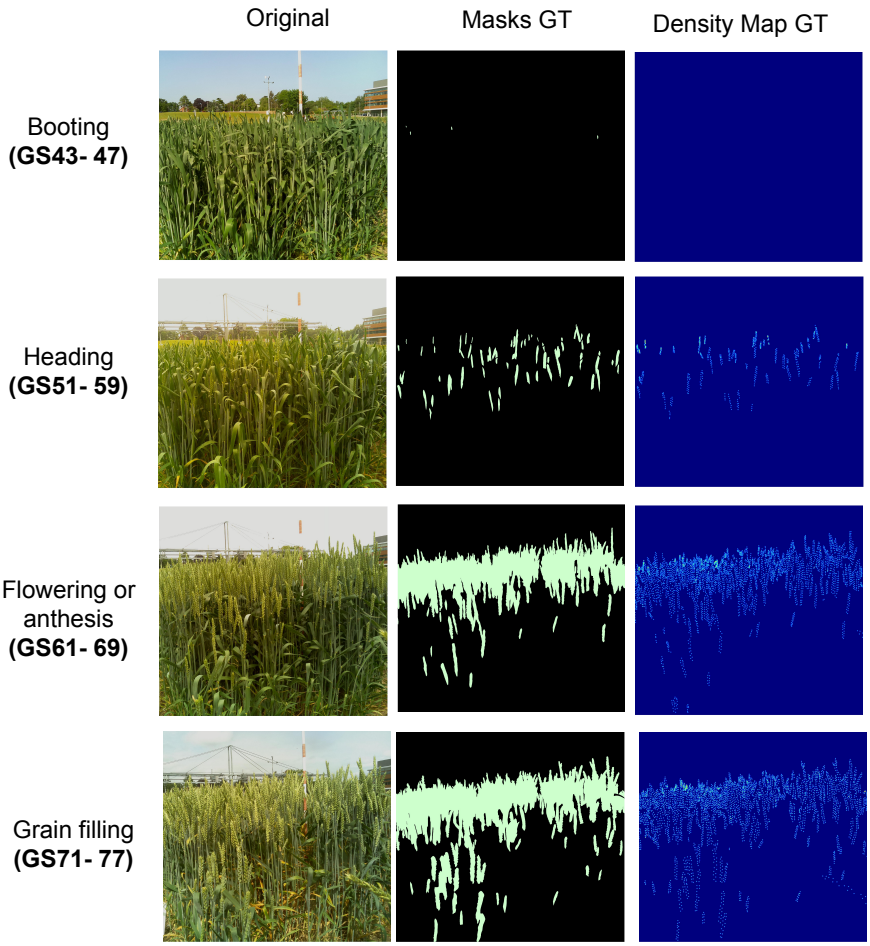


Fig. 3: Wheat growth image series in the field collected by CropQuant workstations of the 2017 growing season, covering booting to grain filling growth stages and showing the corresponding spike regions segmentation masks and density estimation maps

3.5 The Annotated Crop Image Dataset (ACID)

The Annotated Crop Image Dataset (ACID) [34], which is publicly available, has 520 images of wheat plants taken from 21 pots in a glasshouse at a resolution of 1956×1530. Cameras with 12 MP are used to capture the images, which all have a black background. The images represent a variety of spike arrangements and leaves and were taken in uniform lighting. Additionally, dot annotations were added to the images by placing a dot in the center of each spikelet. There are a total of 48,000 spikelets in all images, and the average number of spikelets per scene is 92.3, with a standard deviation of 28.52.

Figure 4 illustrates the similarities and differences between CropQuant and ACID images. For example, all ACID images have a black background and consist-

ent lighting whereas our images can contain a number of objects in the background as well as parts of the sky, variations in lighting, wind conditions, etc. Nevertheless, their dataset of dot annotated wheat images, which is publicly available, may be of use for our own models as the images represent the same physical entity as we are studying and may be used within Transfer Learning.

(a) (b)

Fig. 4: Example of images from (a) ACID dataset and (b) CropQuant dataset

3.6 The protocol for training and validation

For training and testing, we randomly selected sub-images from the original images. We then follow a two-step high-level workflow:

1. Detecting Spike Regions using Semantic Segmentation, in which we first use random sampling to extract patches from the original scene and corresponding mask. Then, we train SpikeSEG [1] with a Fully Convolutional Neural Network to learn non-linear transformations for segmenting spike regions. In the testing phase, a window with the same size as the sampled patch is slid over the test image, and SpikeSEG is used to generate a spike segmentation prediction map.
2. Spikelet Count Estimation Using density estimation, we begin by applying random sampling to extract patches from the original or optimal scene images without background and corresponding dot annotation. Second, we apply a Gaussian smoothing filter ($sigma = 2.5$) to each dot to generate a spikelet density that can be summed to one. Then, for each pixel, we train SpikeCount to perform spikelet density regression. Finally, we perform an integral operation (summation) over the entire map to obtain the spikelet count from a predicted density map.

4 Methods

In this section, we are going to explain the problem statement, SpikeCount architecture, the specification of model training including the cost function, the

Fig. 5: 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

training hyperparameters, and the protocol of splitting the sequences into training, validation and test sets. Again some of this section was presented earlier as part of preliminary published experiments [2] but we include it here for a complete description of our method.

4.1 Problem Statement

We define the problem of spikelet counting mathematically. We model it as a density estimation problem as follows. Given N input images $I_1, I_2, ..., I_N$ with a size of $H \times W \times D$, in our cases representing 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 $P_i = \{P_1, ..., P_{SPC_i}\}$, 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 5(b). To generate the ground truth map GT_i (shown in Fig 5 (c)), a 2D Gaussian kernel $\mathcal{N}(p; P, \sigma^2 1_{2\times 2})$ 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: $GT_i = \{D^{P_1}, ..., D^{P_{H\times W}}\}$ where D^{P_j} is the generated density for the j^{th} pixel in image I_i.

By using the Gaussian kernel, it is possible to reflect the crowding around a spikelet by updating the pixel's density value using information about its neighbours. In other words, the greater the spikelet occlusion in a region, the higher the density values assigned to its pixels.

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

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

4.2 SpikeCount Architecture

To tackle the problem of spikelet counting, we employ a fully convolutional network. Our architecture is represented in Figure 6. The last fully connected layers attached in any CNN-based classifiers are converted to convolutions. This preserves the semantics of target objects, which is critical for tasks requiring structural predictions (predictions for each pixel), such as segmentation, because converting

those layers to convolutions provides information about target objects' location and shape. The FCN model is easily adaptable to address the issue of density estimation because (1) it is a task that requires a structural target (pixel densities) and (2) density estimation can benefit from semantic recognition of the target objects to be counted.

Our model, SpikeCount, is composed of a Very Deep Convolutional Network (VGG16) (Fig. 6 conv1-P5), which is composed of two fully convolutional (Fig. 6 FC6 and FC7) layers and three upsampling layers. All convolutional layers have a filter size of 3 *times* 3 with a stride of 1, while the max-pool layers have a pooling size of 2 *times* 2 with a stride of 2. By adding two skip connections (Fig. 6 after P3 and P4), we can fuse local features associated with spikelets from lower layers with shape and semantic features associated with wheat crops from higher layers. We added upsampling layers to ensure that we recover the original image size that has been reduced due to the application of repetitive convolutions and subsampling.

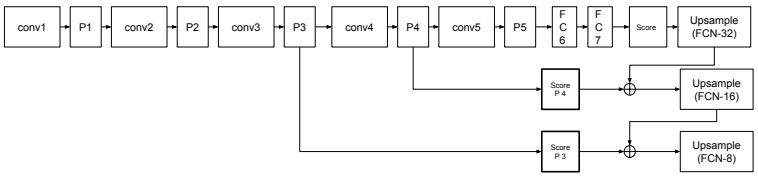


Fig. 6: SpikeCount Architecture established for wheat spikelets density estimation regression.

4.3 SpikeCount Training

4.3.1 Cost Function

We found that using a pixel-wise $L2$ loss function for model optimisation gave the best results to regress the per pixel density (Equation 1) 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:

$$L = \sum_{p \in I_i} (D^p_{GT_i} - D^p_{predicted})^2 \tag{1}$$

4.3.2 Training hyperparameters

Hyperparameters need to be initialised before the training process starts. Then, the training algorithm learns new parameters as part of the learning process [41]. A summary of the SpikeCount training hyperparameters values used in our study follows:

– Weight θ (parameters) /Bias initialisation: We used He et al.'s [18] initialisation. This technique generates a mean centred normal distribution with standard deviation σ equal to $\sqrt{2/n_l}$ where n_l is the number of inputs in a certain layer l.

- Dropout rate probability: We used two dropout layers, with a value of 0.5 for p, which represents the probability of keeping the unit activation or not. This is recommended when using layers with large numbers of units [41] such as FC6 and FC7. Therefore, the dropout is implemented after every fully convolutional layer FC6 and FC7.
- Intermediate non-linearity unit: As a default, we have selected Rectified Linear Unit (ReLu) for this parameter which is an element-wise thresholding operation that is applied on the output of the convolutional layer (resulting feature map) to suppress negative values: F(x)=max(0,x) where x is an element in the feature map.
- Epochs: This refers to the number of training iterations, and is different from one experimental setup to another.
- Optimisation algorithm: The weights are updated for every learning iteration using mini-batch RMSprop optimising algorithm [19], which we selected because it worked better with the density estimation regression problem we have, with a learning rate of 0.001. We have selected the initial learning rate by experimenting with lowest learning rate (1×10^{-5}) and linearly increasing it until we reached the optima, i.e. when we reach the lowest validation loss for mini-batch of 20.

We trained the model using an early stopping technique. Early stopping enables us to keep track of the validation learning for each learning epoch (e.g., cost and accuracy). It is a straightforward and inexpensive method for regularising the model and avoiding overfitting as soon as possible [5, 21]. We have chosen the validation cost as the metric for determining when to stop early. The maximum number of epochs used to observe the change in validation cost is 20. In other words, after 20 epochs, if the validation cost has not decreased, the model training is halted and the model weights with the lowest validation cost are saved.

5 Spikelet Number Estimation set-up

In order to help understand and evaluate the impact of different CNN learning factors that can affect spikelets estimation, we have divided the experimentation into different experimental setups. For all experimental setups, we test SpikeCount on three sets of CQ_2016 and CQ_2017: Original, Optimal and Prediction for 2016 and 2017 image sets respectively. The "Original set" contains images with a complete background, that is, no background is removed, which turned out to be the hardest problem as may be expected. The "Optimal set" refers to images with background removed according to the segmentation ground truth; this scenario enables us to measure the quality of spikelet counting when the background is eliminated completely so we can measure the effect of reducing the complexity of the problem. However, it would require quality segmentation annotations for each image, which may not be available in a "live" crop monitoring scenario. Finally, the "Prediction set" refers to the segmented images when using the prediction of the segmentation model proposed in [1]. This can give us an assessment of the model performance of spikelets estimation relative to the segmentation quality of SpikeCount. It would provide for a completely automated model which could be employed in crop monitoring, but clearly overall quality will be affected by quality of the segmentation model. On the other hand, the removal of the background

even if not perfect would reduce complexity in respect to the Original set. Testing on each set allows us to measure the model performance on specific scenarios.

Our experiments are as follows:

- **Experiment 1: Training SpikeCount from Scratch** - with these set of experiments we try to understand how well the architecture does when trained from scratch in a conventional way.
- **Experiment 2: Training SpikeCount by loading ACID learnt weights** The aim of this experiment is to investigate whether high quality labelled wheat images (ACID images) that are similar to our images but captured in more controlled environment can aid the model learning and improve the overall performance.

  To conduct the experiments described in this paper, we first split the ACID dataset into a training and validation set using the 80:20 split rule. Then, for each set, we randomly sampled sub-images with a size of 512×512. Following that, we manually selected 1241 sub-images containing spike regions from the training set and 303 sub-images containing spike regions from the validation set. For the Transfer Learning experiments, we trained the model on those images for 100 epochs. As described previously, we then loaded parameters learned during model training on the ACID dataset and continued fine tuning the model using the CQ_2016 sequence Original images. For segmented images, we repeated the experiments.

6 Results

After training SpikeCount on CQ_2015 and validating it on CQ_2016 we assess the performance of the model for each experimental set using Original, Prediction and Optimal for both CQ_2016 and CQ_2017. Similar to standard CNN approaches, a form of sliding window (tiling) is used to validate performance on the 2016 and then to test on the 2017 sequence. We scan the target wheat scene sequentially with a window of a size equal to the subsampled patch size selected when training the model with a fixed stride (step size) of s equal to the window size. For each scanned region, the model is applied to predict for spikelet density estimation. Then, the predicted sub-map will be copied in its appropriate location to create a prediction map. The window size corresponds to the sub-image size that is chosen by the experimental setting.

We will analyse the results of spikelet counting in general and in terms of each growth stage as such breakdown may be important from a biological point of view. To analyse the results for each sequence, we need to take the average number of spikelets per growing season into context; these are 2027.50 spikelets for CQ_2016 and 2758.00 spikelets for CQ_2017. We also break down the counting results into background and spike regions for Original and Prediction sets to have a closer look at how the presence of background affects the model's performance.

We evaluate the advantages of Transfer Learning using the additional images in the ACID dataset (Experiment 2) by comparing that to training from scratch (Experiment 1). Transferring knowledge from high quality labelled wheat images (ACID images) that are similar to our images but captured in more controlled environment can aid the model learning and improve the overall performance; the additional annotated images can enhance our training set. Tables 1 and 2 present

the results of testing SpikeCount on CQ_2016 and CQ_2017 respectively. In both tables the results of training SpikeCount on CQ_2015 from scratch and loading ACID learnt weights are compared.

Table 1 corresponding to the 2016 images shows that for the Original set, the model performed better (results highlighted in bold) without the Transfer Learning. When using Transfer Learning weights, the MSE increases by 30% with respect to the values without Transfer Learning and the MAE increases by 26.43%. On the other hand, SMAPE decreased in % value by 34 showing that for this particular measure, performance is better with Transfer Learning.

Table 1: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for two experimental setups (displayed as rows): training SpikeCount from scratch and by loading ACID dataset learned parameters for the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on CQ_2016 images

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**2709.92**	**2566.43**	89.43	1112.86	1015.39	33.58	1904.82	1789.72	64.46
Loading ACID Weights	3864.54	3488.43	**55.43**	**354.57**	**299.44**	**10.44**	**1718.22**	**1590.85**	**47.99**
Abs. Difference	1154.62	922.00	34.00	758.29	715.95	23.14	186.6	198.87	16.47
Improvement as %	29.99	26.43	-	68.14	70.51	-	9.80	11.11	-

Similar to CQ_2016 results, the CQ_2017 results in Table 2 illustrates that for the Original set the model performed better when trained from scratch on measures of MSE and MAE. The MSE decreased by 31.87% and the MAE decreased by 32.25% with respect to values with Transfer Learning. SMAPE, on the other hand, decreased by 33.4% when using Transfer Learning showing an improvement for this particular measure.

Table 2: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for two experimental setups (displayed as rows): training SpikeCount from scratch and by loading ACID dataset learned parameters for Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on CQ_2017 images

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**3747.92**	**3414.08**	93.90	1877.36	1643.33	43.35	2738.24	2461.06	73.67
Loading ACID Weights	5501.65	5039.17	**60.46**	**876.56**	**658.15**	**18.00**	**2252.97**	**1993.42**	**54.49**
Abs. Difference	1753.73	1625.09	33.44	1000.8	985.18	25.35	485.27	467.64	19.18
Improvement as %	31.87	32.25	-	53.3	59.95	-	17.72	19.00	-

To better understand performance, we analysed errors separately for both spike and background regions. This is because the counts may be made of objects that were missed or overestimated in the spike region, but also from objects that were counted in the background region where the count should be zero. On aggregation those may counteract each other so we looked at them separately. We only need to do this for the Original and Prediction sets, since the Optimal set is supposed to remove all the background. For the Original data, we found that the majority of errors were made on the spike regions, though there are a few also in the back-

ground region. However, for Experiment 2 when we apply Transfer Learning the majority of errors are attributable to the background region. Hence, for Original data we can see that the ACID features which contain a very constant background may have a negative impact which translates into maximising the errors on the background regions for our own images with a more complex background. That may explain why Transfer Learning has a negative effect on the Original images for both MSE and MAE measures. The ACID images aid with spike regions because they contain similar detail and features which explains the decrease in SMAPE, but they detract when it comes to counting on the background as they do not add anything for this, having a black background with no objects. This effect is not repeated in the Prediction set where the ratio of errors on the background and spikelet areas is more proportional in both experiments 1 and 2. Note that the Prediction set may have a reduced background area where the segmentation algorithm has not produced particularly good segmentation results, and that in itself would reduce background errors.

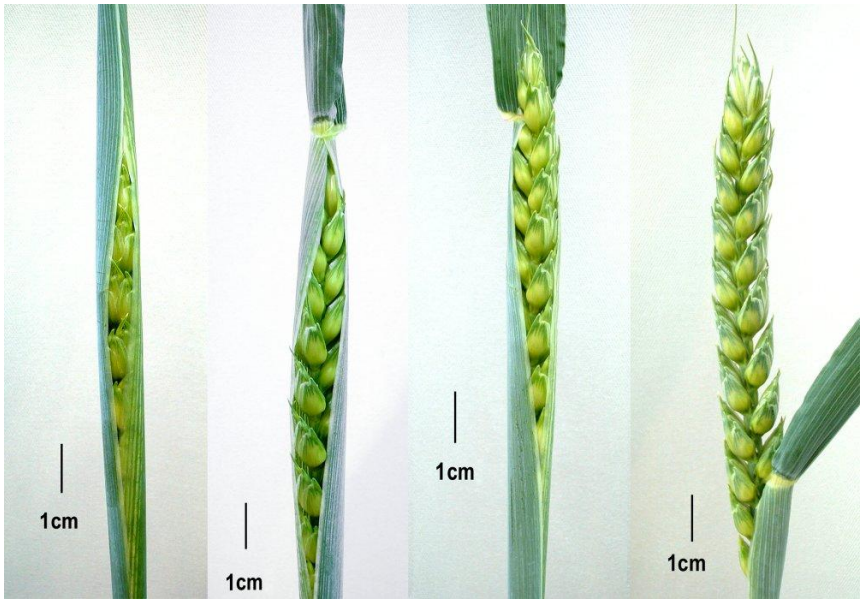
For the Optimal set with no background, according to Table 1 representing CQ_2016, the MSE, and MAE have decreased by 68.14%, 70.51% respectively with respect to values without Transfer Learning and there is also a decrease of 23.14% in SMAPE. Similar reductions of 53.3% for MSE, 59.95% for MAE and 25.35% for SMAPE are seen in Table 2 representing CQ_2017.

Among the three scenarios, eliminating the background completely, as in the Optimal set, has resulted in the best performance for CQ_2016, only missing on average 354.57 spikelets per image for MSE and 299.44 spikelets for MAE and with a relative error of 10.44%. Eliminating the background with our model also produces some gains with respect to the Original images. For CQ_2017, eliminating the background completely has also resulted in the best performance with only missing on average of 876.6 per image for MSE and 658.2 for MAE and with a relative percentage error of 18%. It is clear that focusing the counting on the ROI is important to obtain more accurate counting results.

It is clear also that for both CQ_2016 and CQ_2017 removing the background completely as it is the case with the Optimal set allows the features learned from the ACID dataset through Transfer Learning to have the most marked impact. There is also a positive impact on the Prediction set but not as marked.

6.1 Phenotypic analysis Spikelets Estimation for Growth Stages

In this section, we briefly discuss spikelets estimation results from the perspective of wheat growth stages across the experimental setups. This is important from a biological perspective as growth stages may present very different images with varying numbers of spikelets. As mentioned previously, the wheat images for CQ_2015, CQ_2016 and CQ_2017 were sampled from four different growth stages: booting, Heading, Flowering, and Grain filling. To analyse the results for each growth stage, we need to take the average number of spikelets per growth stage and the number of images into the context.



(a) Late Booting: GS47 **(b)** Late Booting: GS49 **(c)** start of Heading: GS51 **(d)** start of Heading: GS57

Fig. 7: The sequence of spike(ear) emergence at two growth stage Booting and Heading [30, 31]

6.1.1 G40-49: Booting

According to the Zadoka growth decimal scale [49], there are ten major cereal growth stages with each growth stage divided into ten sub-stages [31]. Booting (G40-49) is the fifth growth stage on the Zadoka growth decimal scale. It starts after the stem elongation (G30-39). Booting growth stage starts with visibility of flag leaf sheath (the uppermost leaf on the stem) which contains the spike (ear) inside. So while the ear remains protected by the leaf sheath the stage is booting [30]. Initially, a lower number of spikes may be visible but in late booting they become more partially visible as shown in Figure 7.

There are only two wheat images in the booting growth stage in CQ_2016 with an average of 298 spikelets between them. On the other hand CQ_2017 has five booting images. The last three images of the sequence have an average of 259 spikelets between them. However, for this early growth stage there are no apparent spikes in the first two images of CQ_2017. On inspection, the wheat scenes captured at this growth stage are complex as they are overwhelmed with noisy background objects (i.e leaves) similar in colour and texture to the not so many undeveloped and partially hidden spikes. Therefore, this growth stage will be challenging particularly when testing Original sets.

Table 3 for CQ_2016 and Table 4 for CQ_2017 report on the experimental setups for three sets: Original, Optimal and Prediction. Also, Figures 8 and 9 show visualisation of the spikelet density maps for the Booting stage of 2016/2017 image series.

Table 3: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for the experimental setups (displayed as rows) of the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on **Booting** growth stage of CQ_2016 sequence

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**673.17**	**662.24**	99.80	195.98	187.88	45.42	**555.07**	**543.19**	98.55
Loading ACID Weights	1942.26	1925.73	**99.67**	**179.06**	**177.64**	**45.37**	741.20	719.47	**97.13**

Table 4: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for the experimental setups (displayed as rows) of the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on **Booting** growth stage of CQ_2017 sequence

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**1165.53**	**1072.40**	99.14	82.57	77.73	50.55	407.63	**329.50**	92.39
Loading ACID Weights	5944.61	5832.97	**97.56**	**12.34**	**11.61**	**41.28**	**405.26**	330.83	**89.02**

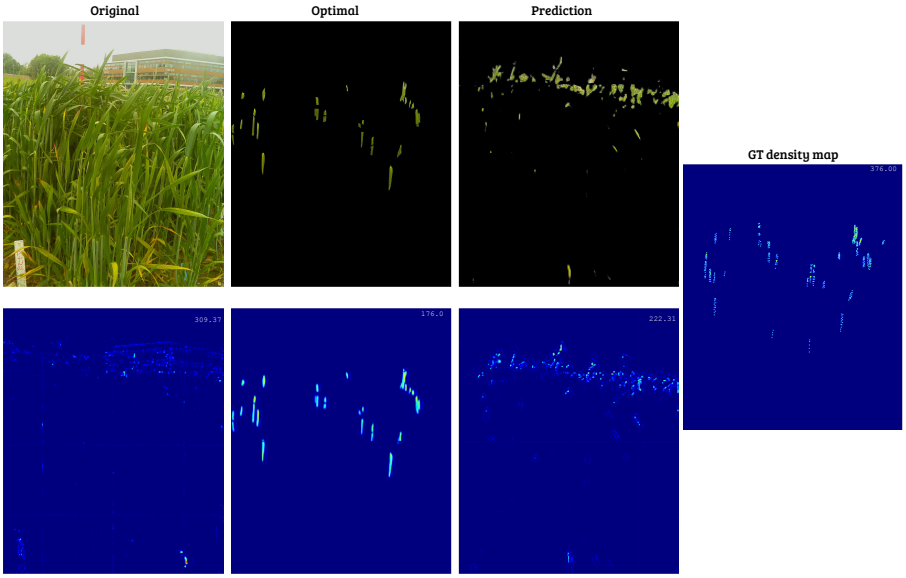


Fig. 8: Visualisation of the spikelets density maps for the **Booting** stage of 2016 image series. The first row shows from left to right: (1) Original image, (2) Optimal image, (3) Prediction image while the second row shows the best resulting density map with predicted spikelets number(top right). For Original and Prediction images the counting is detailed as follows. Top is the total number and bottom is predicted spikelet number within spike region and predicted spikelet number within background when SpikeCount is tested on the corresponding image shown in the first row. The ground truth density map and number of spikelets is presented rightmost.

Booting is a difficult stage to analyse because the background becomes very prominent given the lack of spikelets in images. Therefore best results are obtained when the background is removed completely (Optimal set) as that reduces background errors. This is particularly the case with Transfer Learning as that may help with spikelet identification given additional images from ACID. For the Original set, training from scratch gives best results as that reduces background

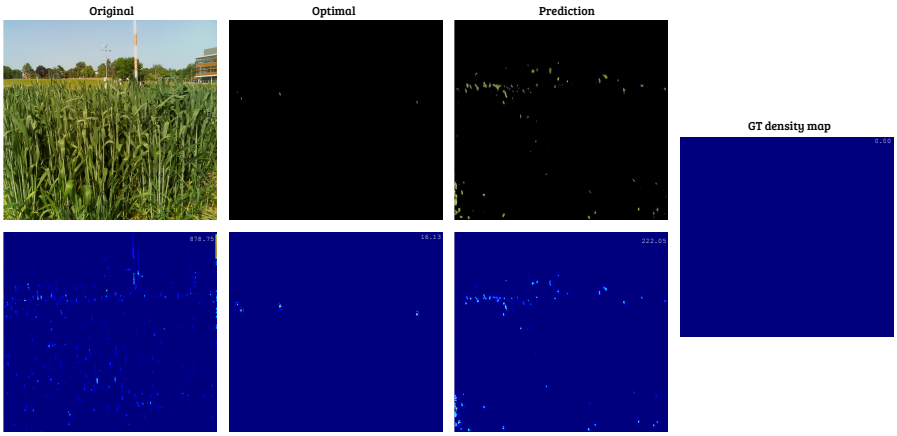


Fig. 9: Visualisation of the spikelets density maps for the **Booting** stage of 2017 image series. The first row shows from left to right: (1) Original image, (2) Optimal image, (3) Prediction image while the second row shows the best resulting density map with predicted spikelets number(top right). For Original and Prediction images the counting is detailed as follows. Top is the total number and bottom is predicted spikelet number within spike region and predicted spikelet number within background when SpikeCount is tested on the corresponding image shown in the first row. The ground truth density map and number of spikelets is presented rightmost.

errors while maintaining spike errors constant. For the Prediction set results are more mixed.

6.1.2 G51-59: Heading

The Heading (ear emergence) is the sixth growth stage according to to the Zadoka growth decimal scale [49]. The Heading stage is considered a key development stage [31] which begins when the ear starts to emerge gradually from the flag leaf sheath until fully emerged [30]. Two key sub stages are important to recognise: half Heading, which is when 50% of the spike has emerged (G55), and full Heading when the full spike has fully emerged (G59). Figure 7 (c) and (d) show two examples of spike emergence in Heading stage.

There are seven wheat images in the Heading growth stage in CQ_2016 with an average of 1464.40 spikelets. For CQ_2017, there are also seven Heading images, which show more crowded spikelet scenes averaging 2970.12 spikelets per image.

The analysis of Heading growth stage for CQ_2016 and CQ_2017 is reported in Table 5 and Table 6 respectively for three sets: Original, Optimal and Prediction. In addition, Figures 10 and 11 show visualisation of the spikelet density maps for the Heading stage of 2016/2017 image series.

For the Heading stage which has more spikelets to count in the images, best results by far are obtained by removing the background (Optimal set) once again. For the Original images, training from scratch gives good results for CQ_2016. For the Prediction set, training from scratch is the best option for CQ_2016 but Transfer Learning has improved the results for CQ_2017 in this stage .

Table 5: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for two experimental setups (displayed as rows) of the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on Heading growth stage of CQ_2016 sequence

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**2212.95**	**2070.73**	88.35	812.22	717.45	31.88	**1718.51**	**1601.91**	72.91
Loading ACID Weights	5135.41	4839.04	**70.46**	**308.93**	**255.21**	**10.606**	1912.97	1806.62	**64.38**

Table 6: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for two experimental setups (displayed as rows) of the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on **Heading** growth stage of CQ_2017 sequence

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**3788.10**	**3515.00**	93.83	2061.41	1827.91	42.21	3024.86	2795.54	77.00
Loading ACID Weights	5757.32	5441.61	**64.95**	**1245.19**	**1063.64**	**19.54**	**2785.21**	**2587.53**	**62.90**

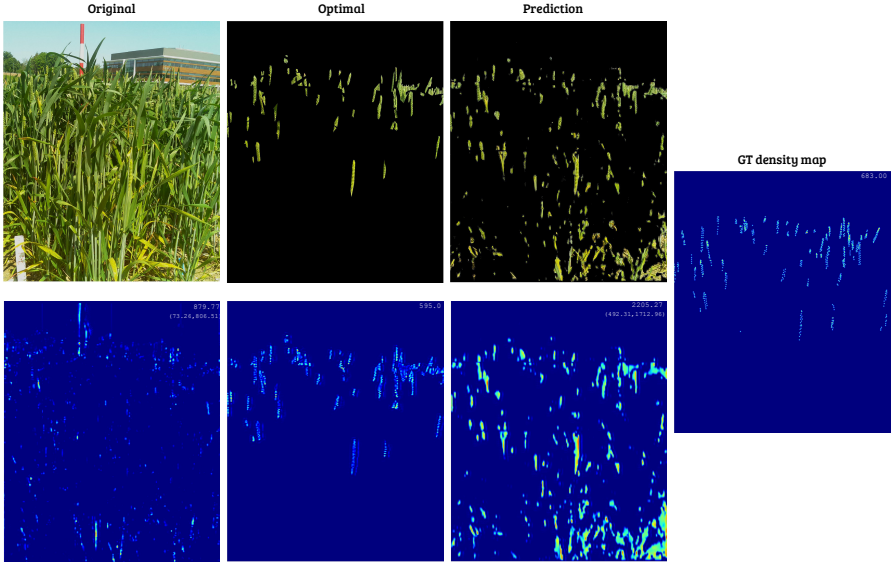


Fig. 10: Visualisation of the spikelets density maps for the **Heading** stage of 2016 image series. The first row shows from left to right: (1) Original image, (2) Optimal image, (3) Prediction image while the second row shows the best resulting density map with predicted spikelets number(top right). For Original and Prediction images the counting is detailed as follows. Top is the total number and bottom is predicted spikelet number within spike region and predicted spikelet number within background when SpikeCount is tested on the corresponding image shown in the first row. The ground truth density map and number of spikelets is presented rightmost.

6.1.3 G61-69: Flowering (anthesis)

The Flowering is the seventh growth stage according to to the Zadoka growth decimal scale [49]. The Flowering stage is considered an key development stage [31] which begins after the Heading stage is complete and can be summarised according to Pask et al. [31] as: “when the 50% of spikes have extruded at least

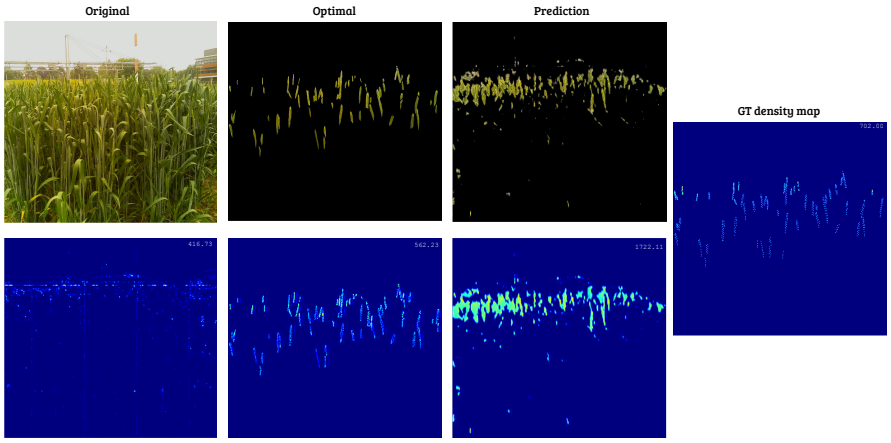


Fig. 11: Visualisation of the spikelets density estimation maps for the **Heading** stage of 2017 image series. The first row shows from left to right: (1) Original image, (2) Optimal image, (3) Prediction image while the second row shows the best resulting density map with predicted spikelets number(top right. For Original and Prediction images the counting is detailed as follows. Top is the total number and bottom is predicted spikelet number within spike region and predicted spikelet number within background when SpikeCount is tested on the corresponding image shown in the first row. The ground truth density map and number of spikelets is presented rightmost.

one anther and [...] mid-anthesis (mid Flowering) is recorded when 50% of spikes have extruded 50% of their anthers".

For this growth stage we have more representatives with 12 wheat images in CQ_2016 with an average of 2244.28 spikelets. For CQ_2017, there are 13 Flowering images, which are more crowded, with spikelets counts averaging 3371.07 per image as was exemplified in Figure 3.

Results for CQ_2016 are presented in Table 7 and for CQ_2017 in Table 8. Again we report each time for three sets: Original, Optimal and Prediction. Figures 12 and 13 show visualisation of the spikelet density maps for the Flowering stage of 2016/2017 image series.

Table 7: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for two experimental setups (displayed as rows) of the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on **Flowering** growth stage of CQ_2016 sequence

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**2753.53**	**2709.59**	88.20	1098.63	1050.07	30.20	1893.45	1846.88	57.43
Loading ACID Weights	3350.92	3066.16	**47.06**	**406.91**	**357.91**	**8.11**	**1656.17**	**1546.81**	**38.85**

For the Flowering growth stage we have more images and high number of spikelets to count. Again, removing the background altogether, as we do for the Optimal set, results in the best errors. For the Optimal set it is using ACID images that results in the best performance in both years. For the Original set, loading ACID weights has not improved the results for both years. For the Prediction set, ACID weights gives the best results. The results of Optimal set are better than the Prediction set results results (Tables 7 and 8).

Table 8: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for two experimental setups (displayed as rows) of the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on **Flowering** growth stage of CQ_2017 sequence

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**3971.32**	**3920.14**	92.19	2061.11	2016.67	42.58	2960.75	2915.79	67.57
Loading ACID Weights	5128.09	4679.23	**50.28**	**831.05**	**713.41**	**11.63**	**2319.48**	**2230.08**	**41.91**

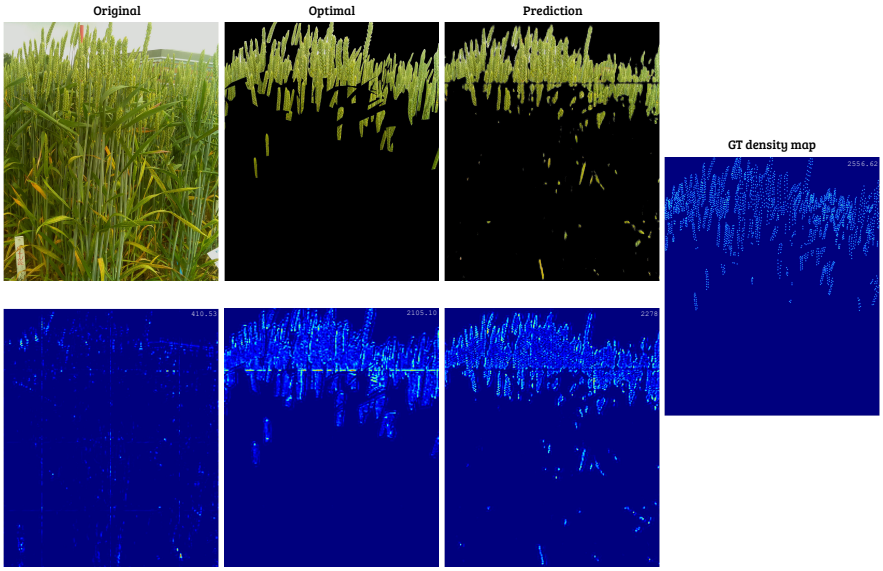


Fig. 12: Visualisation of the spikelets density maps for the **Flowering** stage of 2016 image series. The first row shows from left to right: (1) Original image, (2) Optimal image, (3) Prediction set while the second row shows the best resulted density map with predicted spikelets number(top right). For Original and Prediction images the counting is detailed as follows. Top is the total number and bottom is predicted spikelet number within spike region and predicted spikelet number within background when SpikeCount is tested on the corresponding image shown in the first row. The ground truth density map and number of spikelets is presented rightmost.

6.1.4 GS71-73: Grain filling

The Grain filling is the eighth growth stage according to to the Zadoka growth decimal scale [49]. The Grain filling stage is considered a key development stage [31] which begins after the Flowering stage is complete and is related to the developments of grains [30]. Our dataset for all sequences only contains sub stages from G71 to G73 which are stages when most of Grains contain watery fluids [31]. There are 9 wheat images in the Grain filling growth stage in CQ_2016 with an average of 2560.75 spikelets. For CQ_2017, there are 5 Grain filling images with spikelets counts averaging 3454.62 as shown in Figure 3.

Results for this stage are reported in Table 9 for CQ_2016 and Table 10 for CQ_2017 for three sets: Original, Optimal and Prediction. Also, Figures 14 and 15 show visualisation of the spikelets density maps for the Grain filling stage of 2016/2017 image series.

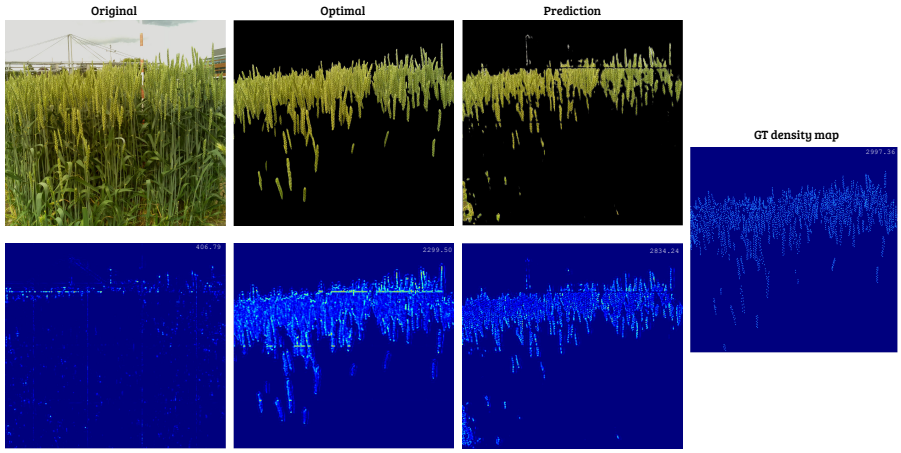


Fig. 13: Visualisation of the spikelets density maps for the **Flowering** stage of 2017 image series. The first row shows from left to right: (1) Original image, (2) Optimal image, (3) Prediction set while the second row shows the best resulted density map with predicted spikelets number (top right). For Original and Prediction images the counting is detailed as follows. Top is the total number and bottom is predicted spikelet number within spike region and predicted spikelet number within background when SpikeCount is tested on the corresponding image shown in the first row. The ground truth density map and number of spikelets is presented rightmost.

Table 9: The Mean Squared Error, Mean Absolute Error, and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for two experimental setups (displayed as rows) of the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on **Grain filling** growth stage of CQ_2016 sequence

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**3201.12**	**3184.25**	89.60	1413.25	1384.76	36.79	2161.81	2136.58	59.69
Loading ACID Weights	3510.72	3348.25	**45.06**	**341.95**	**282.95**	**5.64**	**1724.**	**1675.39**	**36.52**

Table 10: The Mean Squared Error, Mean Absolute Error and Symmetric Mean Absolute Percentage Error of estimating the number of spikelets for two experimental setups (displayed as rows) of the Original, pre-segmented by ground truth (Optimal) and pre-segmented by model (predicted) images (displayed in columns) on **Grain filling** growth stage of CQ_2017 sequence

	Original			Optimal			Prediction		
	MSE	MAE	SMAPE	MSE	MAE	SMAPE	MSE	MAE	SMAPE
Training From Scratch	**4338.18**	**4298.70**	93.22	2036.07	1979.84	39.74	2988.74	2942.03	66.16
Loading ACID Weights	4848.80	4617.84	**43.56**	**802.25**	**593.29**	**9.07**	**2301.58**	**2208.93**	**40.87**

Optimal is as always best as it removes background completely allowing the algorithm to concentrate on the spike area. In this scenario, using ACID weights gives best results. For the Original set, training from scratch has led to the best results. For the Prediction set, using ACID weights gives best results for both growing seasons.

6.2 computational cost

It is not feasible to test the model on a high resolution image (2592×1944) even with a GPU cluster. Therefore, to reduce the computational cost, the testing

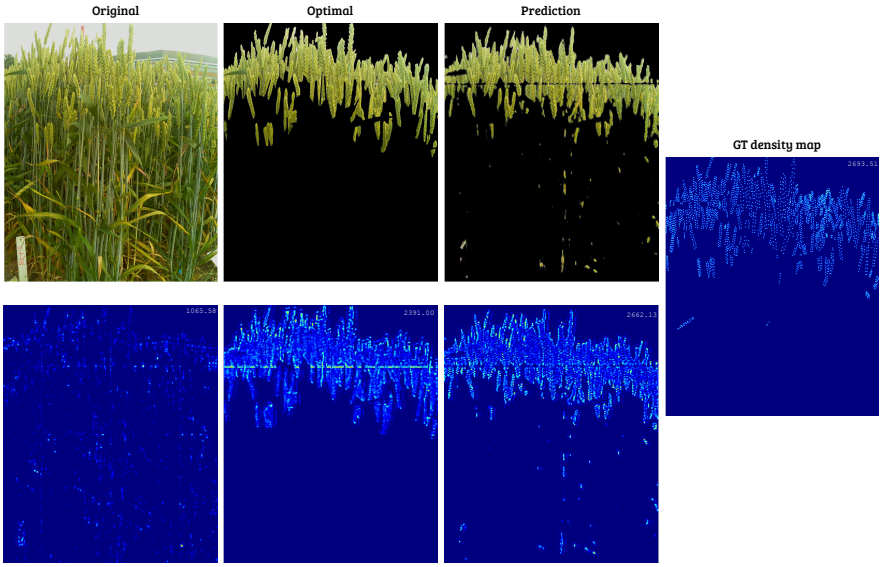


Fig. 14: Visualisation of the spikelets density maps for the **Grain filling** stage of 2016 image series. The first row shows from left to right: (1) Original image, (2) Optimal image, (3) Prediction set while the second row shows the best resulting density map with predicted spikelets number(top right). For Original and Prediction images the counting is detailed as follows. Top is the total number and bottom is predicted spikelet number within spike region and predicted spikelet number within background when SpikeCount is tested on the corresponding image shown in the first row. The ground truth density map and number of spikelets is presented rightmost.

procedure as described previously was achieved using a form of sliding window (tiling). The testing of each high-resolution image took on average less then 10 seconds using the sliding window approach. All experiments were done using High Performance Computing Cluster supported by the Research and Specialist Computing Support service at the University of East Anglia.

7 Conclusions

Counting spikelets from infield wheat crop images is a critical step in quantifying yield traits that can be used to monitor wheat crop growth through automated systems. This is a very challenging task given the variability, self-similarity, varying backgrounds, severe occlusion, density, and changes in illumination etc. associated with spikelets in real wheat images. In this paper, we introduced SpikeCount, a fully convolutional network to count spikelets using a density estimation approach. We have designed several experimental setups to evaluate the model performance such as investigating the effect of transferring learnt features from wheat images captured in controlled environment using Transfer Learning. We have also conducted the evaluation on three variation sets (Original (images with background), Optimal (images with predefined segmentation) and Prediction (the output prediction from the segmentation model discussed in [1])) of two growing seasons 2016 and 2017.

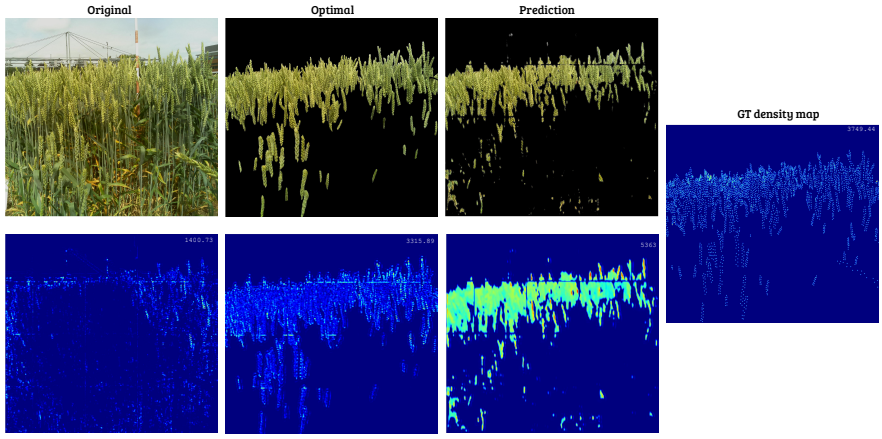


Fig. 15: Visualisation of the spikelets density maps for the **Grain filling** stage of 2017 image series. The first row shows from left to right: (1) Original image, (2) Optimal image, (3) Prediction image while the second row shows the best resulting density map with predicted spikelets number(top right). For Original and Prediction images the counting is detailed as follows. Top is the total number and bottom is predicted spikelet number within spike region and predicted spikelet number within background when SpikeCount is tested on the corresponding image shown in the first row. The ground truth density map and number of spikelets is presented rightmost.

From analysing the result, it is clear that isolating ROI, represented by evaluating on Optimal set, has led to the best results compared to Prediction and Original for all experiments for both CQ_2016 and CQ_2017. The second best results are noted when evaluating on Prediction set. Not removing the background as in Original set has resulted in the worse results which indicate that isolating the ROI (i.e spike regions) plays an important role in solving this problem in this context.

In terms of the best factor for the Optimal set, ACID Transfer Learning has produced best results for CQ_2016 and CQ_2017 which indicates that the transferred features from the ACID parameters have important impact on helping to extract the same traits from images with the same domain but captured in an uncontrolled environment. This may have important implications for other plant phenotyping problems where images captured on controlled or different environments may produce improvements in learning approaches.

Moving to the results of Prediction set, it is clear that employing ACID Transfer Learning has a positive impact for both growing seasons. However, it did not improve the results for the Original set as in this scenario the ACID images may not improve prediction in the background areas, since they do not provide enough information for those, leading to worse results. This shows that learning foreground and background objects may require their own transfer learning examples for best results.

Analysing the results based on growth stages of the Optimal set shows that for more mature wheat images loading ACID weights has led to the best performance for both 2016 and 2017 growing seasons. Regarding the Booting growth stage in CQ_2017, initialising the model with ACID weights and continuing to train

has improved the results of this stage. Finally, transferring ACID knowledge has improve the Heading growth stage results for both seasons.

All in all, it is evident that both eliminating the background and transferring the knowledge learnt from the ACID datasets has a significant effect on counting spikelets from infield uncontrolled images. We conclude that our method, SpikeCount is accurate in the context of very complex infield images and demonstrates the power of deep learning for counting in complex plant phenotyping scenarios.

In further work, we will assess a multi-task learning approach which can leverage the segmentation map learning, for the task of spikelet counting.

8 Declarations

8.1 Funding

Beatriz de la Iglesia received no funds, grants, or other support in relation to this work. Tahani Alkhudaydi was funded for her PhD studies by University of Tabuk.

8.2 Conflicts of interest/Competing interests

The authors have no relevant financial or non-financial interests to disclose.

8.3 Availability of data and material

https://github.com/tanh86/SpikeProject2

8.4 Code availability (software application or custom code)

Not applicable

8.5 Ethics approval

Not applicable

8.6 Consent to participate

Not applicable

8.7 Consent for publication

Not applicable

References

1. Tahani Alkhudaydi, Daniel Reynolds, Simon Griffiths, Ji Zhou, and Beatriz De La Iglesia. An exploration of deep-learning based phenotypic analysis to detect spike regions in field conditions for UK bread wheat. *Plant Phenomics*, 2019:7368761, 2019.
2. Tahani Alkhudaydi, Ji Zhou, and Beatriz de la Iglesia. SpikeletFCN: Counting Spikelets from Infield Wheat Crop Images Using Fully Convolutional Networks. In *International Conference on Artificial Intelligence and Soft Computing*, pages 3–13. Springer, 2019.
3. Carlos Arteta, Victor Lempitsky, J Alison Noble, and Andrew Zisserman. Interactive object counting. In *European Conference on Computer Vision*, pages 504–518. Springer, 2014.
4. Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *European Conference on Computer Vision*, pages 483–498. Springer, 2016.
5. Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.
6. Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.
7. Chen Change Loy, Shaogang Gong, and Tao Xiang. From semi-supervised to transfer counting of crowds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2256–2263, December 2013.
8. Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath RS, Dhruv Batra, and Devi Parikh. Counting everyday objects in everyday scenes. *arXiv preprint arXiv:1604.03505*, 2016.
9. Siu-Yeung Cho, Tommy WS Chow, and Chi-Tat Leung. A neural-based crowd estimation by hybrid global learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(4):535–541, 1999.
10. Joseph Paul Cohen, Genevieve Boucher, Craig A Glastonbury, Henry Z Lo, and Yoshua Bengio. Count-ception: Counting by fully convolutional redundant counting. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 18–26. IEEE, 2017.
11. Jose A. Fernandez-Gallego, Shawn C. Kefauver, Nieves Aparicio Gutiérrez, María Teresa Nieto-Taladriz, and José Luis Araus. Wheat ear counting in-field conditions: high throughput and low-cost approach using rgb images. *Plant Methods*, 14(1):22, Mar 2018.
12. Luca Fiaschi, Ullrich Koethe, Rahul Nair, and Fred A Hamprecht. Learning to count with regression forest and structured labels. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2685–2688. IEEE, 2012.
13. Geoffrey French, Mark Fisher, Michal Mackiewicz, and Coby Needle. Convolutional neural networks for counting fish in fisheries surveillance video. *BMVA Press*, 2015.
14. Geoffrey French, Mark Fisher, Michal Mackiewicz, and Coby Needle. Uea computer vision - image labelling tool. https://bitbucket.org/ueacomputervision/image-labelling-tool, 2015.

15. Mario Valerio Giuffrida, Massimo Minervini, and Sotirios A Tsaftaris. Learning to count leaves in rosette plants. In *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP)*, pages 7–10, 2016.
16. Mehdi Habibzadeh, Adam Krzyżak, and Thomas Fevens. *White Blood Cell Differential Counts Using Convolutional Neural Networks for Low Resolution Images*, pages 263–274. Springer Berlin Heidelberg, 2013.
17. Md Mehedi Hasan, Joshua P. Chopin, Hamid Laga, and Stanley J. Miklavcic. Detection and analysis of wheat spikes using convolutional neural networks. *Plant Methods*, 14(1):100, Nov 2018.
18. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
19. Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8), 2012.
20. Dan Kong, Douglas Gray, and Hai Tao. A viewpoint invariant approach for crowd counting. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1187–1190. IEEE, 2006.
21. Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. Exploring strategies for training deep neural networks. *Journal of machine learning research*, 10(1), 2009.
22. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
23. Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, pages 1324–1332, 2010.
24. Yong Li, Zhengyong Cui, Yingli Ni, Mengjing Zheng, Dongqing Yang, Min Jin, Jin Chen, Zhenlin Wang, and Yanping Yin. Plant density effect on grain number and weight of two winter wheat cultivars at different spikelet and grain positions. *PloS one*, 11(5):e0155351, 2016.
25. Z. Lin and L. S. Davis. Shape-based human detection and segmentation via hierarchical part-template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):604–618, April 2010.
26. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
27. Hao Lu, Zhiguo Cao, Yang Xiao, Bohan Zhuang, and Chunhua Shen. Tasselnet: counting maize tassels in the wild via local counts regression network. *Plant methods*, 13(1):79, 2017.
28. Hao Lu, Liang Liu, Ya-Nan Li, Xiao-Ming Zhao, Xi-Qing Wang, and Zhi-Guo Cao. Tasselnetv3: Explainable plant counting with guided upsampling and background suppression. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2021.
29. Simon Madec, Xiuliang Jin, Hao Lu, Benoit De Solan, Shouyang Liu, Florent Duyme, Emmanuelle Heritier, and Frédéric Baret. Ear density estimation from high resolution rgb imagery using deep learning technique. *Agricultural and Forest Meteorology*, 264:225–234, 2019.
30. University of Bristol. Wheat: The big picture, 2011. Bristol Wheat Genomics.

31. AJD Pask, J Pietragalla, DM Mullan, and MP Reynolds. *Physiological breeding II: a field guide to wheat phenotyping.* CIMMYT, 2012.
32. Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3253–3261, 2015.
33. Dave J. picamera. *Picamera package.* https://picamera.readthedocs.io/en/release-1.13/, 2016.
34. Michael P Pound, Jonathan A Atkinson, Darren M Wells, Tony P Pridmore, and Andrew P French. Deep learning for multi-task plant phenotyping. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 2055–2063. IEEE, 2017.
35. V. Rabaud and S. Belongie. Counting crowded moving objects. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 705–711, June 2006.
36. Mengye Ren and Richard S. Zemel. End-to-end instance segmentation with recurrent attention. In *CVPR*, 2017.
37. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
38. Santi Seguí, Oriol Pujol, and Jordi Vitrià. Learning to count with deep object features. *CoRR*, abs/1505.08082, 2015.
39. Lindsay M Shaw, Adrian S Turner, Laurence Herry, Simon Griffiths, and David A Laurie. Mutant alleles of photoperiod-1 in wheat (triticum aestivum l.) that confer a late flowering phenotype in long days. *PLoS One*, 8(11):e79459, 2013.
40. Lindsay M Shaw, Adrian S Turner, and David A Laurie. The impact of photoperiod insensitive ppd-1a mutations on the photoperiod pathway across the three genomes of hexaploid wheat (triticum aestivum). *The Plant Journal*, 71(1):71–84, 2012.
41. Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.
42. François Tardieu, Llorenc Cabrera-Bosquet, Tony Pridmore, and Malcolm Bennett. Plant phenomics, from sensors to knowledge. *Current Biology*, 27(15):R770–R783, 2017.
43. Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.
44. M. Wang and X. Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *CVPR 2011*, pages 3401–3408, June 2011.
45. Yi Wang and Yuexian Zou. Fast visual object counting via example-based density estimation. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3653–3657. IEEE, 2016.
46. Bo Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 90–97 Vol. 1, Oct 2005.

47. Bolei Xu and Guoping Qiu. Crowd density estimation based on rich features and random projection forest. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.
48. Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.
49. Jan C Zadoks, Ting T Chang, Cal F Konzak, et al. A decimal code for the growth stages of cereals. *Weed research*, 14(6):415–421, 1974.
50. Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015.
51. J Zhou, D Reynolds, T Le Corn, et al. Cropquant: the next-generation automated field phenotyping platform for breeding and digital agriculture. *BioRxiv, Sep*, 2017.
52. Ji Zhou, Daniel Reynolds, ..., and Simon Griffiths. Cropquant: An automated and scalable field phenotyping platform for crop monitoring and trait measurements to facilitate breeding and digital agriculture. *bioRxiv*, 2017.