



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

ScienceDirect



RESEARCH ARTICLE

## Detection and enumeration of wheat grains based on a deep learning method under various scenarios and scales



WU Wei, YANG Tian-le, LI Rui, CHEN Chen, LIU Tao, ZHOU Kai, SUN Cheng-ming, LI Chun-yan, ZHU Xin-kai, GUO Wen-shan

Jiangsu Key Laboratory of Crop Genetics and Physiology/Co-Innovation Center for Modern Production Technology of Grain Crops, Yangzhou University, Yangzhou 225009, P.R.China

### Abstract

Grain number is crucial for analysis of yield components and assessment of effects of cultivation measures. The grain number per spike and thousand-grain weight can be measured by counting grains manually, but it is time-consuming, tedious and error-prone. Previous image processing algorithms cannot work well with different backgrounds and different sizes. This study used deep learning methods to resolve the limitations of traditional image processing algorithms. Wheat grain image datasets were collected in the scenarios of three varieties, six background and two image acquisition devices with different heights, angles and grain numbers, 1748 images in total. All images were processed through color space conversion, image flipping and rotation. The grain was manually annotated, and the datasets were divided into training set, validation set and test set. We used the TensorFlow framework to construct the Faster Region-based Convolutional Neural Network Model. Using the transfer learning method, we optimized the wheat grain detection and enumeration model. The total loss of the model was less than 0.5 and the mean average precision was 0.91. Compared with previous grain counting algorithms, the grain counting error rate of this model was less than 3% and the running time was less than 2 s. The model can be effectively applied under a variety of backgrounds, image sizes, grain sizes, shooting angles, and shooting heights, as well as different levels of grain crowding. It constitutes an effective detection and enumeration tool for wheat grain. This study provides a reference for further grain testing and enumeration applications.

**Keywords:** wheat grain, deep learning, Faster R-CNN, object detection, counting

### 1. Introduction

The number of grains per spike and thousand-grain weight are important factors that reflect the grain yield of crops, and are necessary indicators in crop breeding and cultivation (Prystupa et al. 2004; Peltonen-Sainio et al. 2007; Slafer et al. 2014; García et al. 2016). Enumerating the grains rapidly will render scientific research more efficient. During crop breeding and cultivation, the number of grains per panicle, individual plant, plot, and lineage group is used as the basic information for examining the yield potential

Received 24 April, 2019 Accepted 29 July, 2019  
WU Wei, E-mail: 435208450@qq.com; Correspondence SUN Cheng-ming, Tel: +86-514-879793381, E-mail: cmsun@yzu.edu.cn; GUO Wen-shan, Tel: +86-514-87979339, E-mail: guows@yzu.edu.cn

© 2020 CAAS. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).  
doi: 10.1016/S2095-3119(19)62803-0

of the line (Ferrante et al. 2017), and also constitutes the basic data for examining the effectiveness and practicality of cultivation measures (Li et al. 2004). In practice, the number of grains is often obtained by manual enumeration (Liu et al. 2016). This method is time-consuming, tedious and error-prone. It is thus necessary that a convenient and efficient grain counting method is developed.

At present, commonly used rapid grain detection and counting methods rely on image processing techniques. They have two key processes. The first one is to separate the object from the background, and the other is to resolve the difficulty in separating multiple grains that are adjacent. The typical approach for the first process is to select a single color plate as the background, and then easily separate the object from the background using a color extraction algorithm. The second process focuses on grain counting. Numerous existing segmentation algorithms for adjacent grains have been proposed, including expansion corrosion algorithms (Shatadal et al. 1995), watershed algorithms (Bleau and Leon 2000) and feature point matching algorithms (Hobson et al. 2009; Kiratiratanaaprak and Sinthupinyo 2010; Lin et al. 2014). The threshold of expansion and corrosion is mainly determined by human experience. This is invalid when the grains are adjacent on a large scale, and the non-reproducibility of corrosion and expansion also causes errors. The watershed algorithm is a widely-used segmentation method that uses the local gradient differences in the overlapping regions and identifies the segmentation lines through the canal simulation process. However, this algorithm is prone to over-separation. The feature-point-matching algorithm uses the methods of curvature detection and corner detection to detect the characteristic points of the adjacent grain area, and achieves grain separation through feature point matching (Bai et al. 2009). This method generally has high separation accuracy, but experiences matching errors in complex adhesion areas. Although the accuracy of the existing image processing algorithms is continuously improving, their actual adaptability is still poor because they all require a specific environment, such as a customized background board, a fixed camera, complex equipment, and a fixed angle.

With the rapid development of the theory and practice of deep learning, object detection and classification based on machine learning have entered a new phase. In contrast to the traditional feature extraction algorithms that rely on prior knowledge, deep convolutional neural networks have a certain degree of invariance to geometric transformation, deformation and illumination, which effectively overcomes the difficulty of recognition caused by changes in object appearance. Furthermore, they can also construct feature descriptions that are adaptively driven by training data, thereby exhibiting greater flexibility and generalization ability.

Since the inception of the Region-based Convolutional Neural Network Method (R-CNN) in 2013, deep learning has progressed rapidly. R-CNN, proposed by Girshick et al. (2014, 2015), has made major advances in the field of object detection, and the object detection accuracy has increasingly progressed. Following this, spatial pyramid pooling in deep convolutional networks (SPP-net) (He et al. 2014), Fast Region-based Convolutional Neural Network (Fast R-CNN) (Girshick 2015), Faster Region-based Convolutional Neural Network (Faster R-CNN) (Dechant et al. 2017; Lu et al. 2017; Xiong et al. 2017), region-based fully convolutional networks (R-FCN) (Dai et al. 2016), You Only Look Once (YOLO) (Redmon et al. 2015), Single Shot MultiBox Detector (SSD) (Liu et al. 2015), and other algorithms emerged. These innovative algorithms combined the traditional field of computer vision with deep learning, achieving significant outcomes, such as selective search (SS) and image pyramid. Object detection algorithms based on deep learning have attracted the attention of researchers and have become an important research topic in the field of artificial intelligence. There are more and more studies in the field of agriculture (Liu et al. 2015; Sharada et al. 2016; Chen et al. 2019).

The results of the previous grain counting methods have predominantly been achieved through image processing methods, leaving significant room for improvement in terms of accuracy, scene recognition, calculation speed, and ease of use. Deep learning technology has great advantages in object detection in complex backgrounds, and can improve the deficiencies in past grain enumeration methods. However, wheat grains differ from other objects. There are four or five effective tillers in a single wheat plant, and the number of grains after each tillering is about 40 grains. Thus the scenario of 200 grains appearing on one image at the same time must be considered. This then leads to other issues, e.g., multiple objects and small dimensions on the image. In such a scenario, current deep learning methods cannot be directly used for grain detection and enumeration. Therefore, this study aimed to construct a deep learning-based grain-counting model for better grain counting.

## 2. Materials and methods

We firstly manually collected wheat grain image datasets in different environments, manually annotated the grain, and then divided the dataset into three sub-datasets for model training, verification, and testing, respectively. Model training used the transfer learning method, which can significantly reduce training time and accelerate the reduction of loss. The mean average precision (mAP) is good and smooth during model validation, so the model can be used for

testing. The main procedure is shown in Fig. 1.

## 2.1. Image collection

Publicly available datasets, including ImageNet (<http://www.image-net.org/>), COCO (<http://cocodataset.org/>) and PASCAL VOC ([http://cvlab.postech.ac.kr/~Mooyeol/pascal\\_voc\\_2012/](http://cvlab.postech.ac.kr/~Mooyeol/pascal_voc_2012/)), do not contain the object of this study. Therefore, we manually collected the object dataset. Considering the variability between different wheat varieties, three varieties, namely Ningmai 13, Yangfumai 4 and Yangmai 23, were selected as research objects. In order to adapt the method to a complex scene, six backgrounds were selected: red written, ivory white tablecloth, white A4 paper, black mouse pad, pan-yellow paper, and blue drawing mouse pad, and the number of seeds in each background was 1–220. Differences exist between the front and back images of wheat grains, such as changes in directions of the ventral grooves, or changes in the crowdedness. Different grain spatial arrangements are involved. However, this does not account for changes in grain overlap. Two different image acquisition devices are used including a Huawei mobile phone camera (type: KIW-AL10, camera lens: 13 million pixels f 2.0, China) and a Sony cameras (type: ILCE-6300, lens: E PZ 16–50 mm f 3.5–5.6, Japan). In order to render the method more applicable, three heights (10, 15 and 20 cm) and four shooting angles (45°, 60°, 75°, and 90°) were set for image acquisition. A total of 432 treatments were combined by five treatments: variety,

background, equipment, shooting height, and shooting angle. Each treatment randomly placed 1–200 seeds and captured images. A total of 1 748 images were obtained, including 29 175 grain objects.

## 2.2. Object annotation

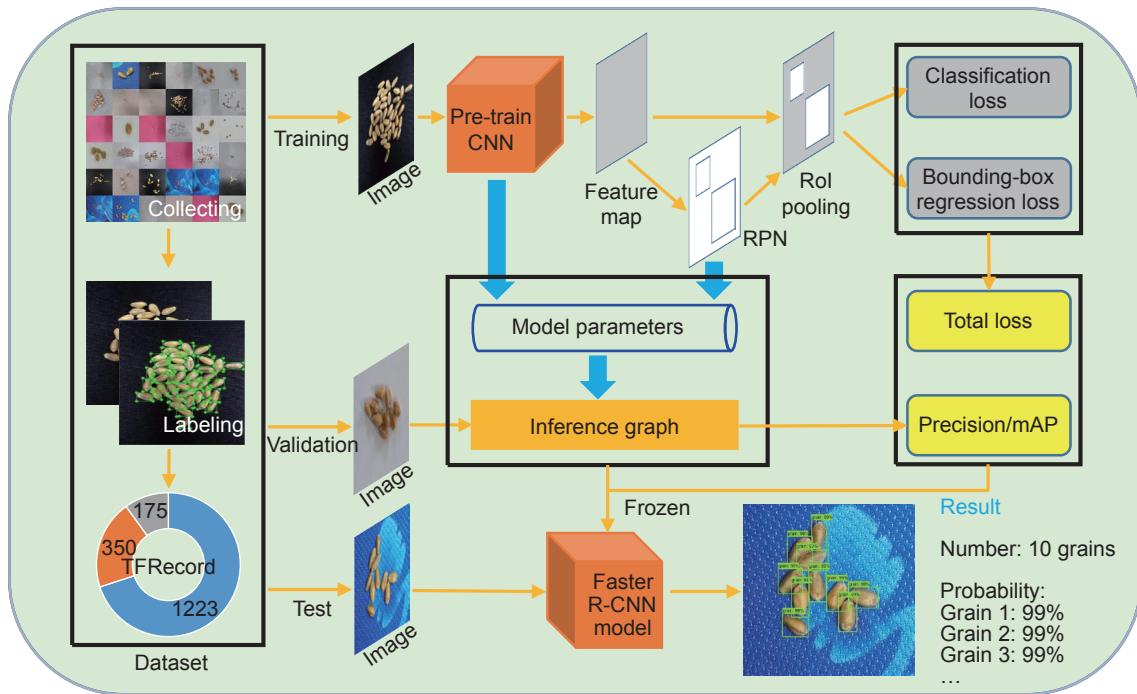
Object detection determines the location of the object. The collected images can be targeted by the object annotation tool. Among the commonly used tools like RectLabel, LabelImg and LabelMe, we selected the LabelImg (Tzutalin, <https://github.com/tzutalin/labelImg>) tool under the Ubuntu System, ultimately generating an .xml file that constitutes a PASCAL VOC format annotation file.

## 2.3. Data standardization

Picture and annotation data were input into the model after being converted into binary data. We used the Tensorflow deep learning framework, and unified the image and annotation data into the TFRecord data type. We divided 70% of the dataset into training sets, 20% into validation sets and 10% into test sets.

## 2.4. Preprocessing

The input images were pre-processed, including the conversion of color space, image turnover, contrast change,



**Fig. 1** Grain detection flowchart. RPN, Region Proposal Network; CNN, Convolutional Neural Network.

and so on. The input of the model is not limited, but different image sizes occupy different amounts of memory or video memory. The maximum image preprocessing size was 1200×1600 pixels, while the minimum was 500×500 pixels.

## 2.5. Network model

The Faster R-CNN Model was used for object detection. Faster R-CNN is developed from R-CNN and Fast R-CNN. Its primary function is to run a convolutional neural network on the original map to extract basic features and introduce a region proposal network (RPN) to create candidate areas, followed by a Softmax classification layer and bounding box return.

Conventional feature extraction networks include VGGNet, Inception Net and ResNet. Deep learning often faces the degradation problem (He and Sun 2014; Srivastava *et al.* 2015), i.e., the error rate increases as the model level deepens. One of the reasons for this is that the deeper the network is, the more the gradient disappears. Therefore, the developers of ResNet (He *et al.* 2016) proposed a residual structure. As the number of layers in the model deepens, this simple structure can resolve the problem of degradation. Fig. 2 shows the ResNet 101 network structure used in this paper.

The Faster R-CNN Model uses RPN networks to extract grain object. The input feature map receives the regional proposal and the regional score through the RPN, uses the non-maximum suppression (threshold 0.7) for the regional score, and then output the Top-N (500 in this article) score

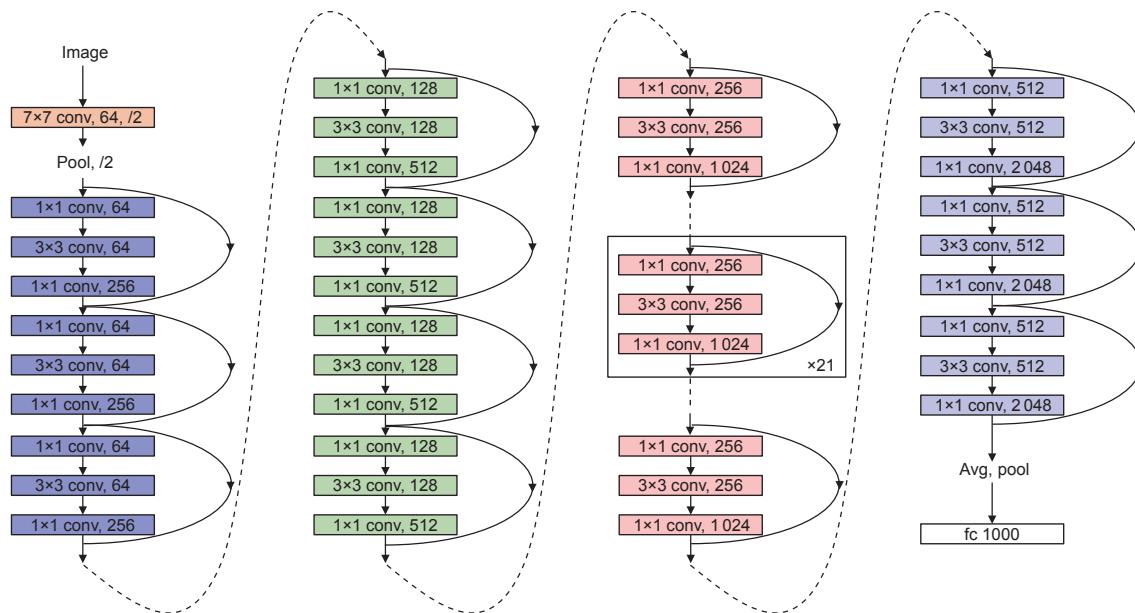
region to the region of interest (RoI) pooling layer. The RPN considers nine possible reference windows at each sliding window position (Fig. 3). Anchors are a fixed set of reference windows with three scales (1 282, 2 562 and 5 122)×three aspect ratios (1:1, 1:2 and 2:1). Using the anchor pyramid to rely solely on single-scale images and feature maps and single-sized convolution kernels, multi-scale problem and multi-length-breadth ratio problem can be solved. This model for the recommended sampling area can achieve good performance in terms of both speed and accuracy.

## 2.6. The assessment method

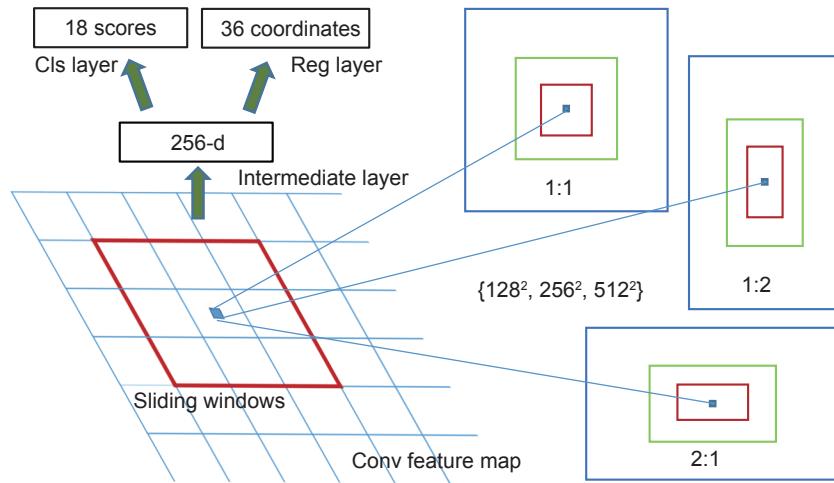
The effect of model fitting is represented by a loss function or cost function. The smaller the loss function is, the better the model fits. The loss function proposed by Faster R-CNN is as follows:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where  $i$  represents the index of an anchor in a batch,  $p_i$  represents the probability of anchor  $i$  predicted as an object; when the anchor is a positive sample,  $p_i^*=1$ , and when it is a negative sample  $p_i^*=0$ . It can be seen that the regression loss term is only activated if the anchor is positive;  $t_i$  represents the four panning scaling parameters of the positive sample anchor to the prediction region;  $t_i^*$  represents the four panning scaling parameters of the positive sample anchor to the ground truth;  $\lambda$  is used to weigh classification loss  $L_{cls}$  and regression loss  $L_{reg}$ ; and the default value of  $\lambda=10$ ;  $N_{cls}$  and  $N_{reg}$  are used to normalize classification loss



**Fig. 2** The ResNet 101 network structure. conv, convolutional.



**Fig. 3** Region Proposal Network (RPN). Cls, classification; Reg, regression; Conv, convolutional.

item  $L_{cls}$  and regression loss item  $L_{reg}$ , respectively.

Classification loss function  $L_{cls}$  is a Boolean classifier (object or not), and the formula is as follows:

$$L_{cls}(p_i, p_i^*) = -\log[p_i^*p_i + (1-p_i^*)(1-p_i)] \quad (2)$$

Regression loss function  $L_{reg}$  is used to calculate the difference between the two transformations, and the formula is as follows:

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (3)$$

R function is defined as:

$$\text{Smooth}_{L1}(x) = \begin{cases} 0.5x^*, & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (4)$$

mAP is an indicator of the accuracy in the object detection algorithm and involves two concepts: precision and recall. For an object, precision and recall can be calculated. A P–R curve can be retrieved through multiple calculations and trials, and the area under the curve is the AP value. The average AP value of each class is mAP. This article only identifies one class, so that the value of mAP lies in the [0, 1] range. The formula is as follows:

$$\text{mAP} = \frac{1}{|Q_R|} \sum_{q \in Q_R} AP(q) \quad (5)$$

where  $Q$  is the number of queries.

Error rates and runtimes are often used as important indicators for measuring count problems. The formula for error rate is:

$$\text{Error ratio} = \frac{PN - TN}{TN} \quad (6)$$

where PN is predicted number, TN is true number.

## 2.7. Computer configuration

The software and hardware of the computer and network configurations are shown in Table 1.

## 3. Results

### 3.1. Image processing

Although the experimental design includes five kinds of treatments of variety, background, equipment, height and angle, the number of images is still small. In order to make the model more adaptable, this study deals with the acquired images, which is beneficial to the expansion of the data set and the increase of sample diversity. The result is shown in Fig. 4, including image rotation, color space conversion and image flipping.

### 3.2. Model training

The model training and verification process need to be monitored. If the training is poor, the parameters should be adjusted accordingly and timely. The model in this article used transfer learning (Pan and Yang 2010) which greatly reduced the training time. The model converged quickly and began to stabilize at 5000 iterations. The classification loss (Fig. 5-A) and the border regression loss (Fig. 5-B) both dropped to a very low level, and the total loss (Fig. 5-C) slowly declined after 5 000–10 000 iterations. The mAP (Fig. 5-D) reached above 0.8 after 5 000–10 000 iterations, and reached 0.91 at 30 000 iterations.

### 3.3. Model accuracy

The model counting method was tested using samples with different grain numbers. The results are shown in Table 2. When the number of grains in the sample is less than 50, the recognition accuracy reaches 100%. When

the number of grains is less than 160, the error rate of most samples does not exceed 1%. When the number of grains reached 200, the error rate increased, but none exceeded 3.5%. Among them, the model is not affected by the type of wheat varieties and different backgrounds, and the robustness is strong.

### 3.4. Model verification

The accuracy of the model was further tested under different backgrounds and different grain numbers. The results are shown in Fig. 6. The results showed that different backgrounds have less impact on the model. The model has strong robust. When the number of seeds is too high, the accuracy of the model will decrease. Fig. 7 is a 1:1 line graph of estimated and measured values. The model performs well. When the number of grains is below 200, the verification model  $R^2$  is 0.9996 and the

RMSE is 2.63.

## 4. Discussion

### 4.1. Number of iterations analysis

The number of iterations had a great influence on the training effect of the model. As shown in Fig. 8, 3000-step iterations were very poor in candidate region selection and category score performance, whereas the results were significantly improved after 30 000 iterations.

### 4.2. Grain scale analysis

Traditional image processing algorithms generally preprocess the image first. When the scale of the grain changes, it is challenging to adaptively adjust the threshold to obtain a better pretreatment effect. As shown in Fig. 9, this research method is barely influenced by the grain scale. Only when the grain scale is extremely small (the grain occupying less than 3% of the entire map) and loses its main features, will the detection results be affected.

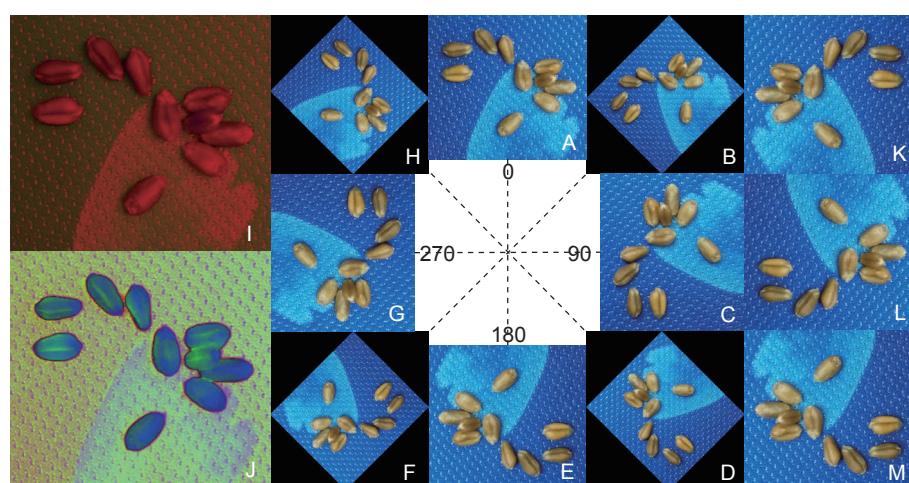
### 4.3. Comparison of grain counting algorithms

From the Table 3, it can be seen that in the traditional grain counting method, the feature point matching algorithm performs best, and the error rate is below 5%. The watershed algorithm is the second, and the morphology erosion algorithm has the highest error rate. Compared with the traditional algorithm, the Faster R-CNN Model has certain advantages in accuracy. When the number of seeds is less than 150, the error rate is less than 1%, and

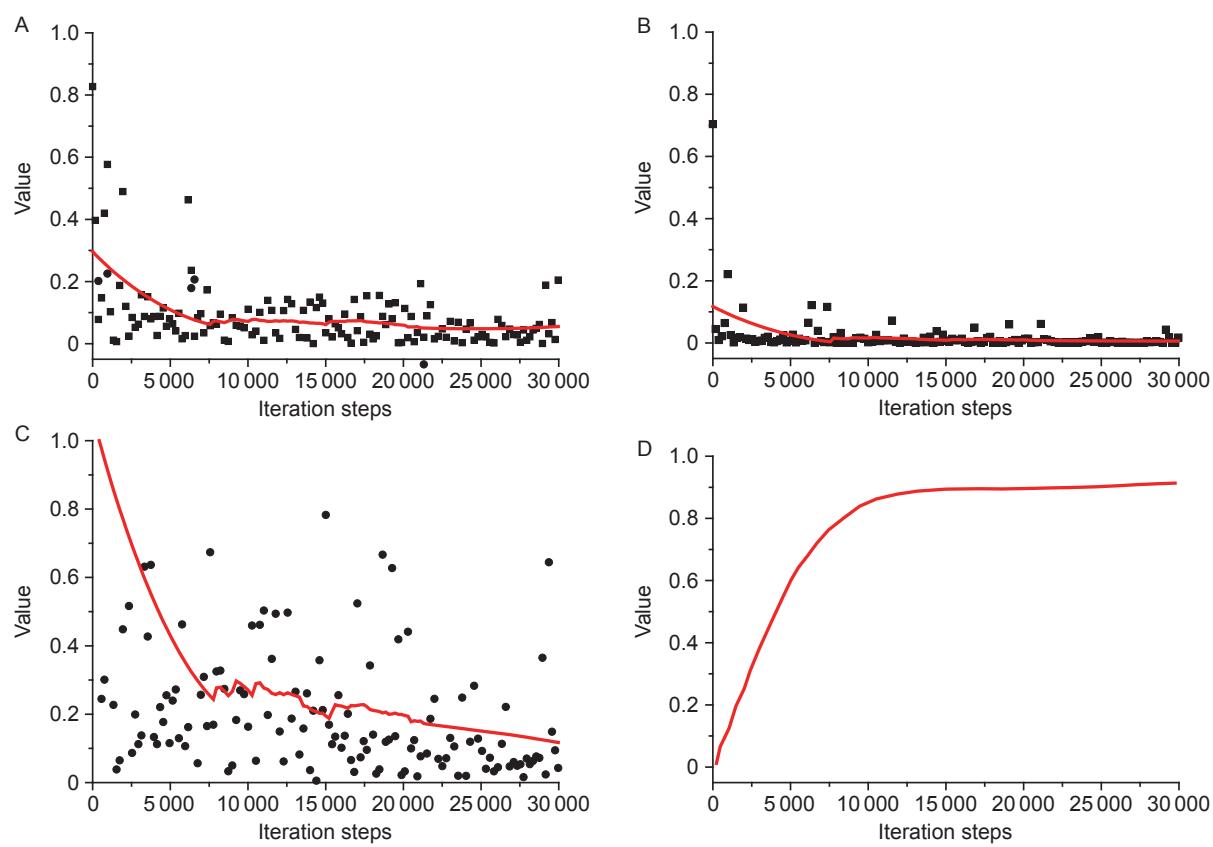
**Table 1** Computer and network configuration

Project <sup>1)</sup>	Content
CPU	Intel Xeon E5-2682v4
RAM	16 G
GPU	Nvidia Tesla P4
Operating system	Ubuntu 16.04 LTS
Cuda	Cuda8.0 with Cudnn v6
Data processing	Python2.7, OpenCV, LabelImg, etc.
Deep learning framework	TensorFlow
Batch size	1
Initial learning rate	0.0003
Learning rate	0.0003
Iteration steps	30 000

<sup>1)</sup> CPU, central processing unit; RAM, random access memory; GPU, graphics processing unit.



**Fig. 4** Image processing results. A, the original image. B, rotation 45°. C, rotation 90°. D, rotation 135°. E, rotation 180°. F, rotation 225°. G, rotation 270°. H, rotation 315°. I, YUV color space. J, HIS color space. K, vertical flip. L, horizontal flip. M, horizontal vertical flip.

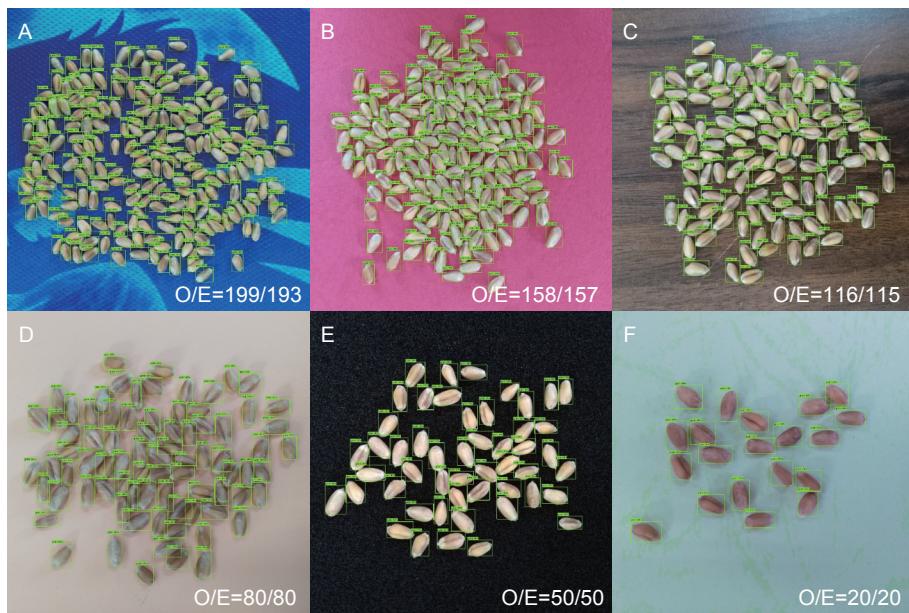


**Fig. 5** Model training and verification results. A, the category prediction loss. B, the border regression loss. C, the total loss. D, mean average precision.

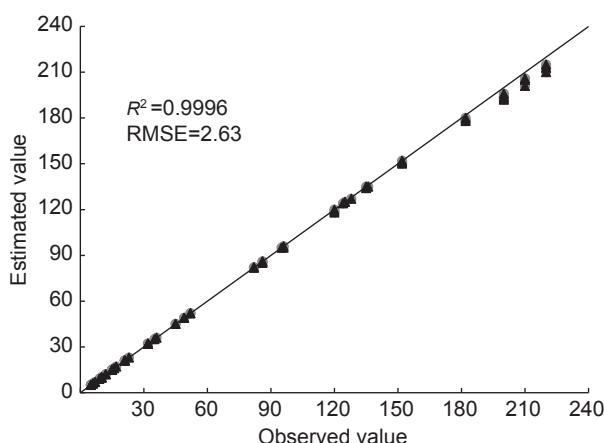
**Table 2** Model counting results under different grain numbers

Variety	Grain number	Error rate (%) <sup>1)</sup>					
		Background 1	Background 2	Background 3	Background 4	Background 5	Background 6
Ningmai 13	10	0	0	0	0	0	0
	20	0	0	0	0	0	0
	50	0	0	0	0	0	0
	80	0.02	0.02	0.03	0.01	0.01	0.02
	120	0.64	0.46	0.81	0.86	0.91	0.67
	160	0.90	0.86	0.95	0.67	0.85	0.89
Yangfumai 4	200	2.67	2.69	3.05	2.87	3.11	3.08
	10	0	0	0	0	0	0
	20	0	0	0	0	0	0
	50	0	0	0	0	0	0
	80	0.01	0.02	0.01	0.02	0.02	0.02
	120	0.54	0.66	0.74	0.69	0.89	0.78
Yangmai 23	160	0.86	0.88	0.96	0.94	1.04	0.99
	200	3.05	2.96	2.58	2.67	2.31	2.77
	10	0	0	0	0	0	0
	20	0	0	0	0	0	0
	50	0	0	0	0	0	0
	80	0.03	0.02	0.01	0.01	0.02	0.02
Yangmai 23	120	0.56	0.84	0.89	0.71	0.85	0.59
	160	0.78	0.93	0.82	0.98	0.97	0.92
	200	2.59	3.10	3.04	2.85	3.06	2.87

<sup>1)</sup> Background 1 is red letter paper, Background 2 is ivory white tablecloth, Background 3 is white A4 paper, Background 4 is black mouse pad, Background 5 is pan-yellow paper, and Background 6 is blue drawing mouse pad.



**Fig. 6** Test results. A, 199 grains in blue drawing mouse pad. B, 158 grains in red letter paper. C, 116 grains in desktop. D, 80 grains in white A4 paper. E, 50 grains in black mouse pad. F, 20 grains in ivory white tablecloth. O, observed value; E, estimated value.

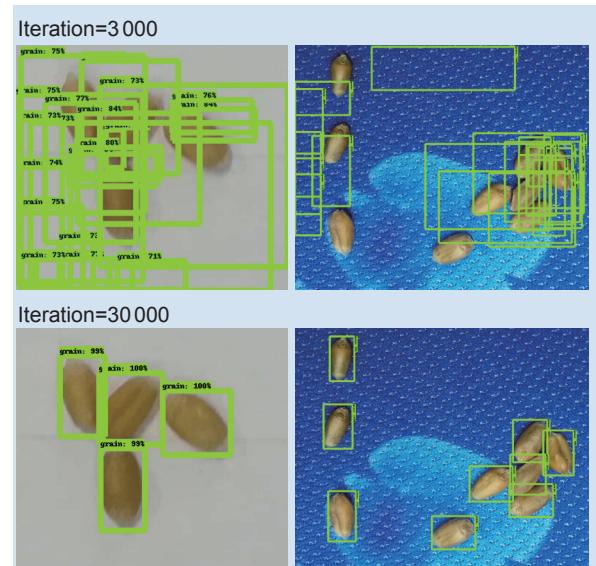


**Fig. 7** Grain counting verification result.

the highest is only about 3%. In the field of deep learning, there are simpler and faster models such as R-FCN, YOLO, SSD, etc. These models perform poorly in the detection of a large number of targets and small targets. The test results represented by SSD are shown in the Table 3. The Faster R-CNN Model is significantly better than them.

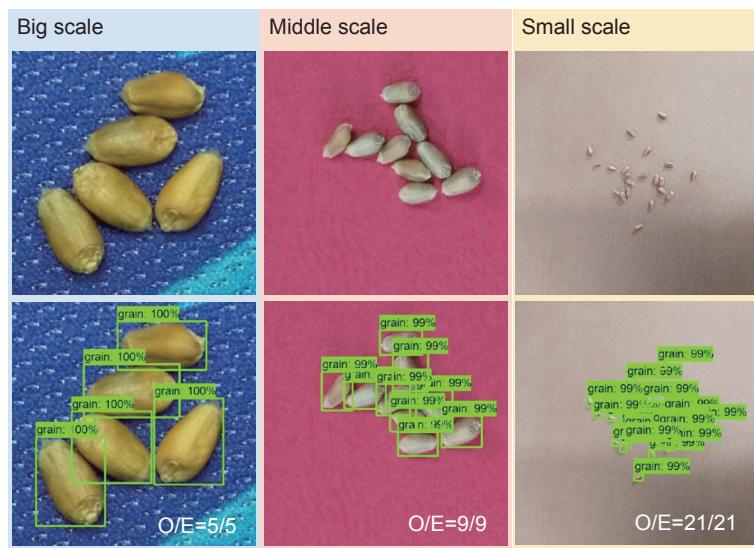
#### 4.4. Runtime

Object detection and enumeration have certain runtime requirements. The optimum is to achieve real-time detection of each frame of the video. Fig. 10 shows a comparison of



**Fig. 8** Results of different numbers of iterations.

several common kernel inspection runtimes. Runtime less than 1 second can be applied in mobile terminal (Liu et al. 2017). The runtime of the expansion corrosion algorithm was less than 1 s, but was less accurate in the case of multi-grain overlapping or adjacency. The watershed algorithm ran for up to 2.6 s. The runtime of the feature-point-matching algorithm increased with the increase in grain number. None of these three algorithms could segment and recognize



**Fig. 9** Test results of different grain scales. O, observed value; E, estimated value.

**Table 3** Comparison of error rates in the different grain counting methods<sup>1)</sup>

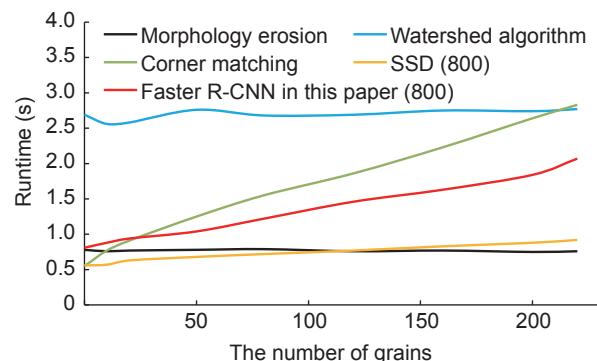
Grain number	Morphology erosion (%)	Watershed algorithm (%)	Corner matching (%)	SSD (%)	Faster R-CNN in this paper (%)
10	10.32	18.62	0	0	0
20	17.65	23.46	0	0	0
50	32.23	35.64	1.24	10.56	0
80	36.26	38.26	3.56	25.43	0
120	36.56	39.35	2.46	33.34	0.84
160	42.48	38.46	2.78	37.50	0.63
200	45.56	37.59	4.32	48.36	3.02

<sup>1)</sup> SSD, single shot multibox detector; Faster R-CNN, Faster Region-based Convolutional Neural Network.

other background and size images. The SSD Model was insensitive to changes in grain number and image size (Figs. 10 and 11), and the runtime was less than 1 s. The runtime of the Faster R-CNN Model in this paper increased with the increase in grain number, and was within 2 s. The larger image size also affected the runtime of the Faster R-CNN Model. Once the image size was larger than 800×800 pixels, it had no effect on the detection accuracy of the model. Therefore, it is possible to resize the image to 800×800 pixels to resolve this issue. Below 500×500 pixels, too many object features will be lost, and the accuracy of both models will be impaired. The accuracy of the Faster R-CNN Model was about 0.2 higher than that of the SSD Model. Overall, the Faster R-CNN model was superior. Future work should be focused on optimizing the model structure and reducing the model parameters, which will help reduce the runtime.

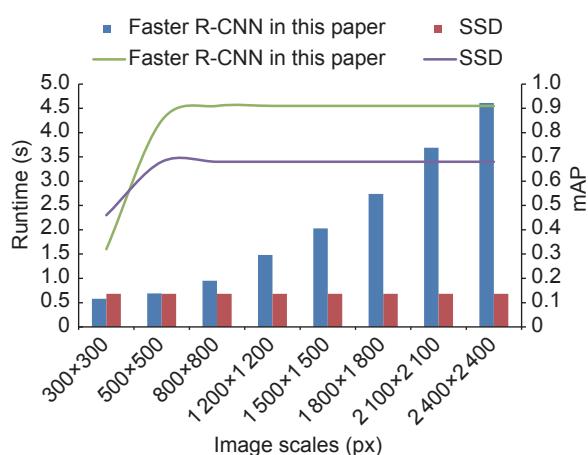
## 5. Conclusion

This paper collected a wheat grain dataset and used the



**Fig. 10** Comparison of the runtime of the different methods. SSD, single shot multibox detector; Faster R-CNN, Faster Region-based Convolutional Neural Network. 800, the size of the image is 800×800 pixels.

migration learning method based on the Faster R-CNN deep learning network to train the wheat grain detection and enumeration depth mode. The model can be applied to the detection and enumeration of images in a variety of complex backgrounds at multiple scales, and in multi-angle



**Fig. 11** Effect of different image scales on runtime. SSD, single shot multibox detector; Faster R-CNN, Faster Region-based Convolutional Neural Network. px, pixels.

environments. With an error rate of less than 3% and a running time less than 2 s, the model constitutes an effective tool for the detection and enumeration of wheat grains.

## Acknowledgements

This research was mainly supported by the National Key Research and Development Program of China (2017YFD0301205), the Postgraduate Research & Practice Innovation Program of Jiangsu Province, China (KYCX18\_2371), the National Natural Science Foundation of China (31701355 and 31671615), the China Postdoctoral Science Foundation, China (2016M600448), the Priority Academic Program Development of Jiangsu Higher Education Institutions, China (PAPD), the Yangzhou Science Foundation for Excellent Youths, China (YZ2017098) and the Science and Technology Plan Projects of Yangzhou, China (YZ2016251).

## References

- Bai X, Sun C, Zhou F. 2009. Splitting touching cells based on concave points and ellipse fitting. *Pattern Recognition*, **42**, 2434–2446.
- Bleau A, Leon L J. 2000. Watershed-based segmentation and region merging. *Computer Vision & Image Understanding*, **77**, 317–370.
- Chen J, Liu Q, Gao L W. 2019. Visual tea leaf disease recognition using a convolutional neural network model. *Symmetry*, **11**, 343–356.
- Dai J, Li Y, He K, Sun J. 2016. R-FCN: Object detection via region-based fully convolutional networks. Neural Information Processing Systems (NIPS). Barcelona, Spain. pp. 1–9.
- Dechant C, Wiesnerhanks T, Chen S, Stewart E L, Yosinski J, Gore M A, Nelson R J, Lipson H. 2017. Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology*, **107**, 1426–1432.
- Ferrante A, Cartelle J, Savin R, Slafer G A. 2017. Yield determination, interplay between major components and yield stability in a traditional and a contemporary wheat across a wide range of environments. *Field Crops Research*, **203**, 114–127.
- García G A, Serrago R A, Drecer M F, Miralles D J. 2016. Post-anthesis warm nights reduce grain weight in field-grown wheat and barley. *Field Crops Research*, **195**, 50–59.
- Girshick R. 2015. Fast R-CNN. In: *IEEE International Conference on Computer Vision*. IEEE Computer Society, USA. pp. 1440–1448.
- Girshick R, Donahue J, Darrell T, Malik J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, USA. pp. 580–587.
- Girshick R, Donahue J, Darrell T, Malik J. 2015. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **38**, 142–158.
- He K, Sun J. 2015. Convolutional neural networks at constrained time cost. In: *Computer Vision and Pattern Recognition*. IEEE Computer Society, USA. pp. 5353–5360.
- He K, Zhang X, Ren S, Sun J. 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **37**, 1904–1916.
- He K, Zhang X, Ren S, Sun J. 2016. Deep residual learning for image recognition. In: *Computer Vision and Pattern Recognition*. IEEE Computer Society, USA. pp. 770–778.
- Hobson D M, Carter R M, Yan Y. 2009. Rule based concave curvature segmentation for touching rice grains in binary digital images. In: *Instrumentation and Measurement Technology Conference, 2009. I2MTC 09*. IEEE Computer Society, USA. pp. 1685–1689.
- Kiratiratanapruk K, Sinthupinyo W. 2010. Segmentation algorithm for touching round grain image. In: *International Conference on Electronics and Information Engineering*. IEEE, Japan. pp. V1-263–V1-266.
- Li J, Thomson M, McCouch S R. 2004. Fine mapping of a grain-weight quantitative trait locus in the pericentromeric region of rice chromosome 3. *Genetics*, **168**, 2187–2195.
- Lin P, Chen Y M, He Y, Hu G W. 2014. A novel matching algorithm for splitting touching rice kernels based on contour curvature analysis. *Computers & Electronics in Agriculture*, **109**, 124–133.
- Liu T, Chen W, Wang Y, Wu W, Sun C, Ding J, Guo W. 2017. Rice and wheat grain counting method and software development based on Android system. *Computers and Electronics in Agriculture*, **141**, 302–309.
- Liu T, Wu W, Chen W, Sun C, Chen C, Wang R, Zhu X, Guo W. 2016. A shadow-based method to calculate the percentage

- of filled rice grains. *Biosystems Engineering*, **150**, 79–88.
- Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, Berg A C. 2015. SSD: Single Shot MultiBox Detector. ECCV, Netherlands. pp. 21–37.
- Lu J, Hu J, Zhao G, Mei F, Zhang C. 2017. An in-field automatic wheat disease diagnosis system. *Computers & Electronics in Agriculture*, **142**, 369–379.
- Pan S J, Yang Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge & Data Engineering*, **22**, 1345–1359.
- Peltonen-Sainio P, Kangas A, Salo Y, Jauhainen L. 2007. Grain number dominates grain weight in temperate cereal yield determination: Evidence based on 30 years of multi-location trials. *Field Crops Research*, **100**, 179–188.
- Prystupa P, Savin R, Slafer G A. 2004. Grain number and its relationship with dry matter, N and P in the spikes at heading in response to N×P fertilization in barley. *Field Crops Research*, **90**, 245–254.
- Redmon J, Divvala S, Girshick R, Farhadi A. 2015. You only look once: Unified, real-time object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*.
- IEEE Computer Society, USA. pp. 779–788.
- Sharada P M, David H, Marcel S. 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, **7**, 1419–1429.
- Shatadal P, Jayas D S, Bulley N R. 1995. Digital image analysis for software separation and classification of touching grains. I. Disconnect algorithm. *Transactions of the ASAE*, **38**, 645–649.
- Slafer G A, Savin R, Sadras V O. 2014. Coarse and fine regulation of wheat yield components in response to genotype and environment. *Field Crops Research*, **157**, 71–83.
- Srivastava R K, Greff K, Schmidhuber J. 2015. Highway networks. In: *International Conference on Machine Learning*. ICML, France. pp. 1–6.
- Xiong X, Duan L, Liu L, Tu H, Yang P, Wu D, Chen G, Xiong L, Yang W, Liu Q. 2017. Panicle-SEG: A robust image segmentation method for rice panicles in the field based on deep learning and superpixel optimization. *Plant Methods*, **13**, 104–119.

Executive Editor-in-Chief LI Shao-kun  
Managing editor WANG Ning