An integrated rice panicle phenotyping method based on X-ray and RGB scanning and deep learning

Lejun Yu, Jiawei Shi, Chenglong Huang, Lingfeng Duan, Di Wu, Debao Fu, Changyin Wu, Lizhong Xiong, Wanneng Yang, Qian Liu

PII: S2214-5141(20)30109-4
DOI: https://doi.org/10.1016/j.cj.2020.06.009
Reference: CJ 503

To appear in: The Crop Journal

Received date: 23 February 2020
Revised date: 24 May 2020
Accepted date: 30 June 2020


This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier.
An integrated rice panicle phenotyping method based on X-ray and RGB scanning and deep learning

Lejun Yu\textsuperscript{a,1}, Jiawei Shi\textsuperscript{b,1}, Chenglong Huang\textsuperscript{b}, Lingfeng Duan\textsuperscript{b}, Di Wu\textsuperscript{b}, Debao Fu\textsuperscript{b}, Changyin Wu\textsuperscript{b}, Lizhong Xiong\textsuperscript{b}, Wanneng Yang\textsuperscript{b, *}, Qian Liu\textsuperscript{c,a, *}

\textsuperscript{a}Britton Chance Center for Biomedical Photonics, Wuhan National Laboratory for Optoelectronics, and Key Laboratory of Ministry of Education for Biomedical Photonics, Department of Biomedical Engineering, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China
\textsuperscript{b}National Key Laboratory of Crop Genetic Improvement, National Center of Plant Gene Research, Agricultural Bioinformatics Key Laboratory of Hubei Province, and College of Engineering, Huazhong Agricultural University, Wuhan 430070, Hubei, China
\textsuperscript{c}School of biomedical engineering, Hainan University, Haikou 570228, Hainan, China

Abstract: Rice panicle phenotyping is required in rice breeding for high yield and grain quality. To fully evaluate spikelet and kernel traits without threshing and hulling, using X-ray and RGB scanning, we developed an integrated rice panicle phenotyping system and a corresponding image analysis pipeline. We compared five methods of counting spikelets and found that Faster R-CNN achieved high accuracy ($R^2$ of 0.99) and speed. Faster R-CNN was also applied to indica and japonica classification and achieved 91% accuracy. The proposed integrated panicle phenotyping method offers benefit for rice functional genetics and breeding.

Keywords: Rice (\textit{O. sativa}); Panicle traits; RGB imaging; X-ray scanning; Faster R-CNN

1. Introduction

Rice (\textit{Oryza sativa} L.) is one of the most important food crops worldwide [1]. With the continued growth of the world population [2], the production of grain has become particularly important, leading to an urgent need for breeding high-yield rice. Rice yield depends on three traits: number of panicles, number of spikelets per panicle, and kernel weight [3], and the number of spikelets is positively and strongly associated with the number of primary branches (PBs) per panicle [4]. The size and shape of kernels affect yield and market value [5]. The growth of rice panicles can also be used to evaluate disease status [6], nutrients [7], and growth period [8]. For this reason, phenotyping of rice panicles and kernels is necessary for rice research. However, conventional measurement of these traits is time-consuming and labor-intensive. Novel and effective techniques to evaluate panicle traits are urgently needed.
In recent years, several photonics-based methods have been developed to evaluate grain traits. A yield trait scorer (YTS) has been developed that achieves fully automatic rice threshing and measures yield-related traits with a MAPE of less than 5% and processes 1440 plants over 24 working hours per day [9]. A software program, GrainScan, has been developed [10] for measuring kernel size and color. However, these methods require threshing or shelling before inspection, increasing the time cost of trait extraction and preventing the measurement of panicle-structure traits. A method has been developed to identify sprouted and healthy kernels using soft X-ray scanning and neural networks [11]. Neethirajan et al. [12] applied a dual energy X-ray imaging technique to classify vitreous and non-vitreous kernels with accuracy reaching 93%. A new method has been proposed [13] to distinguish between filled and unfilled spikelets by combining visible light imaging and soft X-ray imaging. These X-ray imaging methods can measure the inner structure of seeds without shelling. However, threshing is still needed, and panicle-structure traits still cannot be evaluated.

Without threshing, efforts using machine vision have been made [14–15] to evaluate panicle traits. Huang et al. [16] presented a rice panicle length-measuring system based on dual-camera imaging, with measurement accuracy and efficiency reaching 92.8% and 900 panicles per hour, respectively. Zhou et al. [17] developed a semi-automated phenotyping pipeline to evaluate sorghum panicle traits for subsequent genome-wide association studies (GWAS). A panicle trait phenotyping tool (P-TRAP) [18] performed high-throughput panicle image analysis and evaluated panicle and spikelet traits. A novel image-analysis pipeline, PANorama [19], measured panicle traits and showed potential to shed light on rice panicle architecture. These tools provide software support for evaluating panicle and grain traits. However, tedious manual intervention is necessary in some parts of image processing. For example, with P-TRAP, manual correction is required after image processing and spikelet number calculation, raising the time and labor costs.

In recent years, deep learning applications have grown rapidly in agriculture [20]. One example is the use of convolutional neural networks (CNN) [21] to estimate numbers of soybean seeds and count fruits [22]. With the development of deep learning, Faster R-CNN [23] has been widely used in crop organ and disease detection. Liu et al. [24] used Faster R-CNN to identify maize tassels in RGB images acquired by unmanned aerial vehicle (UAV) with accuracy of up to 95.0%. Wang et al. combined Faster R-CNN with several deep CNNs to rapidly and accurately identify tomato diseases [25]. Wu et al. [26] used the transfer learning method combined with Faster R-CNN to count wheat kernels and calculate relevant traits in a variety of scenarios, with average accuracy reaching 0.91. To distinguish corn from weeds and improve the accuracy of corn yield prediction, Xia et al. [27] designed an end-to-end solution using an improved Faster R-CNN, greatly increasing the accuracy and speed of
differentiation. Quan et al. [28] described a Faster R-CNN method based on pre-trained network VGG19 that achieved 97.7% precision in differentiating maize seedlings from soil and weeds. Jin et al. [29] combined Faster R-CNN with the region growth algorithm to successfully segment maize stems.

We describe a bimodal imaging method for rice panicles without threshing and hulling, using RGB and X-ray scanning and an image analysis pipeline to evaluate spikelet and kernel shape and number.

2. Materials and methods

2.1. Rice materials

Forty rice accessions, mainly from China, Southeast Asia, and South Asia with a small number from Africa and the Americas (Fig. 1a), including 20 indica and 20 japonica accessions, were randomly selected from 529 O. sativa accessions [30]. All accessions were sown with five replicates at Huazhong Agriculture University (Wuhan City, Hubei province, China; 30.5°N, 114.3°E) in a field plot (Fig. 1b) and harvested in October 2016 for phenotyping. For each accession, five panicles from the main stems (a total of 200 panicles) were randomly selected for further imaging and analysis.

Fig. 1 – Global distribution of 40 rice accessions and field experiment.
(a) The star represents the growing site of the 40 groups of accessions (Wuhan city, Hubei province, China; 30.5N, 114.3E), and the red dots represent the origins of the 40 rice accessions. (b) The field experiments.
2.2. RGB image acquisition

The RGB imaging platform was a miniature photo studio that contains a smartphone, a backlight, and a calibration scale (Fig. 2a). A Huawei Mate8 smartphone (NXT-AL10/3GB RAM) was employed to acquire panicle images. The smartphone was fixed to the top of the platform with double-sided adhesive tape and a backlight was placed on the bottom of the platform as the light source. Before imaging, branches were separated with tweezers and each panicle was affixed to a sheet of white paper. A barcode (tag) for each sample was located in the upper left corner of the paper and a calibration ruler was placed on the right side of the paper. The resolution of the original image was 4608 × 3456 pixels and the image was clipped to 2550 × 1800 pixels to remove irrelevant background for subsequent processing (Fig. 2b) using OpenCV 4.0 and Python 3.6. The RGB images were processed to evaluate spikelet number, spikelet morphology, and panicle structure traits (Fig. 2f). The average time cost of RGB image acquisition (including branch separation, panicle fixation, and image acquisition) was ~5 min.

---

**Fig. 2 – Flowchart of RGB, X-ray image acquisition and processing.**
(a) The RGB imaging platform. (b) The RGB image. (c) The X-ray imaging platform. (d) The X-ray image. (e) Image analysis pipeline. (f) Panicle traits.
2.3. X-ray image acquisition

In our previous study [31], a micro-computed tomography (CT) system was developed to evaluate rice tiller traits. In the present study, the micro-CT imaging system was improved to screen panicles nondestructively. Figure 2c shows the system, which consists of an X-ray source (Nova 600, Oxford, UK), an X-ray flat panel detector (1536 × 1920, PaxScan, VARIAN, USA), and a linear stage controlled by a programmable logic controller (PLC) (CP1H, OMRON, Japan). The power of the X-ray source was set as 40 kV and 40 W. Because the panicle was larger than the field of view of the X-ray detector, the linear stage was added to move the panicle forward and backward to acquire two images (1536 × 1920 pixels) at two positions. The two images were then merged, as shown in Fig. 2d. The X-ray images were further processed to extract kernel number and shape traits.

2.4. Spikelet shape extraction using RGB panicle images

An integrated image analysis pipeline consisting of three projects was developed. The projects were a deep learning project for spikelet counting, a LabVIEW (National Instruments Corporation, USA) project for spikelet shape detection, and another LabVIEW project for kernel shape determination and kernel counting. P-TRAP [18] was also used to process panicle RGB images and determine spikelet number and panicle shape.

An image analysis pipeline was developed using LabVIEW to extract spikelet shape traits from RGB panicle images. (1) First, the R component of each image was extracted (Fig. 3b) and threshold segmentation was applied to yield a binary image (Fig. 3c). (2) Open operation was then performed, with the main aim of removing most of the branches (Fig. 3d and e). (3) Small areas were removed to yield images without impurities and noise (Fig. 3f). (4) The barcode (tag) located near the image border, which contained small perforations, was detected and deleted (Fig. 3h). (5) The remaining main stem was detected and removed, having a larger elongation factor than the other objects (Fig. 3j). (6) Finally, an appropriate area threshold and elongation factor segmentation was adopted to identify non-adhering spikelets to calculate spikelet shape. All of the spikelet traits are described in Table 1. A video of the operating procedure of the LabVIEW project for spikelet shape detection (Supplementary Video 1.mp4) and the source code (directory: spikelet-grain-traits_extraction) can be downloaded from http://plantphenomics.hzau.edu.cn/data/The%20Crop%20Journal-Supplementary%20File.rar or https://pan.baidu.com/s/14qbsSH76Y6jrDnLBinN_w (Extraction code: qcts).
Fig. 3 – Flow chart of spikelet shape trait extraction based on RGB image.
(a) Original RGB image. (b) Grayscale image. (c) Image after binarization. (d) Image after open operation. (e) Branch in the enlarged image. (f) Image after removal of small objects. (g) Tag in enlarged image. (h) The image after deletion of the tag area. (i) The main stem in the enlarged image. (j) The image after removal of the main stem. (k) A single spikelet in the enlarged image.
<table>
<thead>
<tr>
<th>Trait classification</th>
<th>Traits</th>
<th>Abbreviation</th>
<th>Mean ± SD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spikelet traits</strong></td>
<td>Spikelet number</td>
<td>SN</td>
<td>190.09 ± 45.58</td>
</tr>
<tr>
<td></td>
<td>Spikelet length (mm)</td>
<td>SL</td>
<td>8.63 ± 1.15</td>
</tr>
<tr>
<td></td>
<td>SD of the spikelet length</td>
<td>SD_SL</td>
<td>0.70 ± 0.34</td>
</tr>
<tr>
<td></td>
<td>Spikelet width (mm)</td>
<td>SW</td>
<td>2.92 ± 0.39</td>
</tr>
<tr>
<td></td>
<td>SD of the spikelet width</td>
<td>SD_SW</td>
<td>0.36 ± 0.18</td>
</tr>
<tr>
<td></td>
<td>Spikelet length/width ratio</td>
<td>SLWR</td>
<td>0.41 ± 0.08</td>
</tr>
<tr>
<td></td>
<td>SD of the spikelet length/width ratio</td>
<td>SD_SLWR</td>
<td>0.07 ± 0.05</td>
</tr>
<tr>
<td></td>
<td>Spikelet area (mm²)</td>
<td>SA</td>
<td>17.36 ± 2.81</td>
</tr>
<tr>
<td></td>
<td>SD of the spikelet area</td>
<td>SD_SA</td>
<td>2.31 ± 1.29</td>
</tr>
<tr>
<td></td>
<td>Spikelet perimeter (mm)</td>
<td>SP</td>
<td>19.84 ± 2.26</td>
</tr>
<tr>
<td></td>
<td>SD of the spikelet perimeter</td>
<td>SD_SP</td>
<td>1.63 ± 0.84</td>
</tr>
<tr>
<td></td>
<td>Spikelet area/perimeter ratio</td>
<td>SA_PP</td>
<td>0.87 ± 0.10</td>
</tr>
<tr>
<td></td>
<td>SD of the spikelet area/perimeter ratio</td>
<td>SL_SA_R</td>
<td>0.09 ± 0.06</td>
</tr>
<tr>
<td></td>
<td>Spikelet equivalent ellipse</td>
<td>SL_EE</td>
<td>3.41 ± 0.70</td>
</tr>
<tr>
<td></td>
<td>SD of the spikelet equivalent ellipse</td>
<td>SD_SSEE</td>
<td>0.69 ± 0.69</td>
</tr>
<tr>
<td></td>
<td>Spikelet circularity</td>
<td>SC</td>
<td>1.36 ± 0.11</td>
</tr>
<tr>
<td></td>
<td>SD of the spikelet circularity</td>
<td>SD_SC</td>
<td>0.10 ± 0.09</td>
</tr>
<tr>
<td><strong>Kernel traits</strong></td>
<td>Kernel number</td>
<td>KN</td>
<td>136.82 ± 42.26</td>
</tr>
<tr>
<td></td>
<td>Kernel length (mm)</td>
<td>KL</td>
<td>6.22 ± 1.31</td>
</tr>
<tr>
<td></td>
<td>SD of the kernel length</td>
<td>SD_KL</td>
<td>0.52 ± 0.37</td>
</tr>
<tr>
<td></td>
<td>Kernel width (mm)</td>
<td>KW</td>
<td>2.45 ± 0.55</td>
</tr>
<tr>
<td></td>
<td>SD of the kernel width</td>
<td>SD_KW</td>
<td>0.23 ± 0.17</td>
</tr>
<tr>
<td></td>
<td>Kernel length/width ratio</td>
<td>KLWR</td>
<td>0.36 ± 0.11</td>
</tr>
<tr>
<td></td>
<td>SD of the kernel length/width ratio</td>
<td>SD_KLWR</td>
<td>0.07 ± 0.15</td>
</tr>
<tr>
<td></td>
<td>Kernel area (mm²)</td>
<td>KA</td>
<td>11.67 ± 2.99</td>
</tr>
<tr>
<td></td>
<td>SD of the kernel area</td>
<td>SD_KA</td>
<td>1.52 ± 0.87</td>
</tr>
<tr>
<td></td>
<td>Kernel perimeter (mm)</td>
<td>KP</td>
<td>14.66 ± 2.88</td>
</tr>
<tr>
<td></td>
<td>SD of the kernel perimeter</td>
<td>SD_KP</td>
<td>1.11 ± 0.84</td>
</tr>
<tr>
<td></td>
<td>Kernel area/perimeter ratio</td>
<td>KAPR</td>
<td>0.77 ± 0.17</td>
</tr>
<tr>
<td></td>
<td>SD of the kernel area/perimeter ratio</td>
<td>SD_KAPR</td>
<td>0.06 ± 0.05</td>
</tr>
<tr>
<td></td>
<td>Kernel equivalent ellipse</td>
<td>KEE</td>
<td>2.72 ± 0.75</td>
</tr>
<tr>
<td></td>
<td>SD of the kernel equivalent ellipse</td>
<td>SD_KEE</td>
<td>0.51 ± 0.95</td>
</tr>
<tr>
<td></td>
<td>Kernel circularity</td>
<td>KC</td>
<td>1.22 ± 0.22</td>
</tr>
<tr>
<td></td>
<td>SD of the kernel circularity</td>
<td>SD_KC</td>
<td>0.07 ± 0.11</td>
</tr>
<tr>
<td><strong>Panicle structure traits</strong></td>
<td>Primary axis length (mm)</td>
<td>PL</td>
<td>174.98 ± 32.97</td>
</tr>
<tr>
<td></td>
<td>Primary axis diameter (mm)</td>
<td>PD</td>
<td>1.90 ± 0.54</td>
</tr>
<tr>
<td></td>
<td>Node number</td>
<td>NN</td>
<td>10.60 ± 2.03</td>
</tr>
<tr>
<td></td>
<td>Primary branch number</td>
<td>PBN</td>
<td>14.49 ± 2.69</td>
</tr>
<tr>
<td></td>
<td>Primary branch length average (mm)</td>
<td>PBLA</td>
<td>93.24 ± 14.93</td>
</tr>
<tr>
<td></td>
<td>Primary branch intervals length (mm)</td>
<td>PBI</td>
<td>13.27 ± 2.81</td>
</tr>
<tr>
<td></td>
<td>Secondary branch number</td>
<td>SBN</td>
<td>36.72 ± 10.63</td>
</tr>
<tr>
<td></td>
<td>Secondary branch length average (mm)</td>
<td>SBLA</td>
<td>23.16 ± 4.30</td>
</tr>
<tr>
<td></td>
<td>Secondary branch intervals length (mm)</td>
<td>SBI</td>
<td>6.26 ± 3.07</td>
</tr>
</tbody>
</table>
### 2.5. Spikelet number evaluation with and without deep learning using RGB panicle images

To achieve fully automated calculation of the spikelet number of one panicle, five methods (with and without deep learning) were applied and compared:

1. **Area method:** After Fig. 3j (including adhering and non-adhering spikelets) and Fig. 3k (non-adhering spikelets) were obtained, we selected the non-adhering spikelets (Fig. 3k) to calculate the average area of single spikelets (AASS) and the total spikelet number $N_{\text{AREA}}$:

   $$N_{\text{AREA}} = \frac{\text{TSA}}{\text{AASS}}$$

   where TSA (Total Spikelet Area) is the total area of both the adhering and non-adhering spikelets (Fig. 3j).

2. **P-TRAP:** We also applied P-TRAP [18] to process the panicle RGB images (Fig. 3a) to obtain the number of spikelets, $N_{\text{P-TRAP}}$.

The third to fifth methods involved deep learning techniques for our spikelet counting task. Its core was to identify all the spikelets in the image, a task requiring accurate object detection models. The algorithms used in deep learning for object detection can be divided into two types. One is the two-stage method, which divides the detection process into two parts to generate candidate boxes and classify objects within candidate boxes. The other is the one-stage method, which unifies the process and rapidly and directly provides detection results. By contrast, the main feature of the two-stage method is high precision but much lower speed.

3. **Faster region-based convolutional neural network (Faster R-CNN) method (two-stage):** The structure of Faster R-CNN is shown in Fig. 4. First, the whole image after resizing (1333 × 800 pixels) is input into the Resnet101-based [32] CNN for feature extraction. We adopted the structure of the feature pyramid network (FPN) [33] to improve the detection effect of small targets by using multiscale feature maps. The full feature maps are then sent to the region proposal network (RPN) [34], which generates multiple anchor boxes with different aspect ratios in each feature map for searching possible regions of interest (ROI). There are two sibling branches in RPN for predicting bounding boxes. The classifier can determine whether anchors belong to the foreground or the background and the regressor can fix the coordinates of anchors. Through the RPN, the corresponding location can be extracted from the generated feature maps as the proposed regions required by the RoIAlign layer [35], so
that each ROI can generate a fixed-size feature map. Finally, we get the classification probability and box regression value using a series of fully connected layers (FCs).

![Diagram of deep learning method process](image)

*Fig. 4 – Process flow of the deep learning method.*

(4) Cascade R-CNN method (two-stage): To achieve more accurate object detection, Cascade R-CNN [36] was developed. Cai et al. [36] found that the higher the training intersection-over-union (IOU) threshold was set, the higher the accuracy of the predicted boxes would be. However, once threshold training or testing is improved, detection does not improve as much as might be expected. A higher IOU threshold results in many missed detections. To solve this problem, Cascade R-CNN employs a cascading approach, with the output of the previous detection model as the input of the subsequent detection model and the IOU threshold of the latter being higher. This ensures the detection of objects but also improves the accuracy of the prediction box.

(5) Single Shot MultiBox Detector (SSD) method (one-stage): For the one-stage method, we chose the mainstream SSD [37] model to experiment. The backbone is VGG-16 [38]. The main feature of SSD is its high speed, but it also uses six different feature maps to detect objects at different scales to achieve a higher precision.

Data annotation was implemented with the open-source label toolkit LabelImg [38]. The RGB images (Fig. 2b) after clipping were used for data annotation (Fig. 5). The dataset partition is summarized in Table 2. The experiment was based on the Ubuntu operating system, and the graphics processing unit (GPU) was an NVIDIA 2080 (8 GB graphic memory), while the central processing unit (CPU) was an Intel Core i5-8500 (32 GB memory). The models was built, trained, and verified using the Python language. Based on the PyTorch deep learning framework, computed unified device architecture (CUDA) version 10.1 was used for the parallel computing framework. The major hyperparameter setting included non maximum suppression (NMS) threshold of
0.74, batch size of 2, learning rate of 0.02, and maximum epoch number of 1000, which were specified in a config file we supplied. All the deep learning methods were trained and tested in the open-source object detection framework MMDetection [39], which can be downloaded from https://github.com/open-mmlab/mmdetection, and the overall process steps are described in the documentation (README.docx) we provide. The documentation, our trained weights (weight_spikelet_number.pth) for spikelet counting, the video of the operation procedure of the deep learning project for spikelet counting (Supplementary Video 2.mp4), and the source code of the deep learning project (directory: spikelet_number) for spikelet counting can be downloaded from http://plantphenomics.hzau.edu.cn/data/The%20Crop%20Journal-Supplementary%20File.rar or https://pan.baidu.com/s/14qbsSH76Y6jrDnLBinN_w (Extraction code: qcts).

Fig. 5 – Labeled image using LabelImg.

<table>
<thead>
<tr>
<th>Table 2 – Image (sample) numbers of the dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Train</td>
</tr>
<tr>
<td>Test</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

2.6. Accuracy evaluation of five methods for spikelet counting

To evaluate the measurement accuracy of the five methods for spikelet counting, 80 test images were used. LabelImg was used to obtain the ground truth by labeling the spikelets in these 80 test images. The true spikelet number was then obtained from the generated annotation file and called value $N_{\text{manual}}$. The advantage of this
method is it was necessary only to mark all the spikelets in each image instead of counting them at the same time, effectively reducing manual measurement error.

\( R^2 \), mean absolute percentage error (MAPE), and root mean square error (RMSE) values were used to evaluate the accuracy of each methods. The specific equations are as follows:

\[
\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{|N_{\text{prediction}} - N_{\text{manual}}|}{N_{\text{manual}}} \right) \times 100\%
\]  

(2)

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (N_{\text{prediction}} - N_{\text{manual}})^2}
\]  

(3)

where \( N_{\text{prediction}} \) is the number predicted with the five methods.

### 2.7. Panicle structure trait extraction using P-TRAP

With the obtained RGB images, P-TRAP, developed by AL-Tam et al. [18] was used to extract the panicle structure traits and spikelet number \( N_{\text{P-TRAP}} \) (listed in Table 1). After the user defines the starting point and the ending point, this Java-based tool can automatically find the nodes of the panicle and calculate panicle structure traits.

### 2.8. Kernel number and kernel shape extraction using X-ray images

For each panicle, two original X-ray scanning images (Fig. 6a) were obtained and a LabVIEW project was developed to extract the kernel number and shape, as shown in Fig. 6. (1) First, clicking on the same position in two images (shown with red points in Fig. 4a), allowed merging the two original X-ray scanning images (Fig. 6a) to obtain a complete X-ray image of the panicle (Fig. 6b). (2) Gray filtering was performed to increase image contrast and remove noise (Fig. 6c). (3) After filtering, an adaptive threshold was set to perform threshold segmentation (Fig. 6d). (4) After segmentation, a small area was removed (Fig. 6e) to yield a binary image leaving only rice kernels (Fig. 6f). (5) There are some adhering kernels in Fig. 6g. To accurately obtain the kernel number, a distance transformation method was adopted to disconnect the adhering kernels (Fig. 6i) and calculated the kernel number (Fig. 6h). (6) To obtain the morphological properties of the kernels, the image before distance transformation was used (Fig. 6f), an appropriate area threshold was set to delete the adhering kernels, and an image with only the non-adhering kernels (Fig. 6j and k) was used to calculate the kernel shape. The time consumption of X-ray image analysis for one sample was ~20 s. A video of the operation procedure of the image analysis pipeline (Supplementary Video 3.mp4) and the source code of the LabVIEW project for kernel trait extraction (directory: spikelet-grain-trait-extraction) can be downloaded from
Fig. 6 – Flowchart of kernel shape trait extraction based on X-ray image.
(a) The original X-ray image. (b) The image after merging. (c) The image after filtering. (d) The image after binarization. (e) Small objects in enlarged image. (f) The image after removing small objects. (g) Adhering kernels in enlarged image. (h) The image after distance transformation. (i) The disconnection of adhering kernels in the enlarged image. (j) The image after removing the adhering kernels. (k) A single kernel in the enlarged image.

2.9. Three types of spikelet classification method

After 45 digital traits were acquired, three types of methods were used to distinguish indica and japonica accessions, and the classification results were examined and compared.

(1) Support vector machine (SVM). The SVM is a classic supervised machine learning method [40] widely used in statistical classification and regression analysis. The main idea of SVM is to establish a classification hyperplane as the decision surface to maximize the difference between positive and negative samples. In this study, the Scikit-learn package [41] was used to implement the SVM model. For training data, the 45 acquired traits were used and divided into three categories, RGB, X-ray, and all traits, to evaluate the classification effect of these three categories. Recursive feature elimination (RFE) [42] was used to calculate the contribution of each trait to the SVM model.

(2) Image classification deep learning network. A pre-trained deep learning model called Resnet50 [32] was used, this is a class of deep neural networks composed of 50 layers in total, most commonly applied in visual
analysis. In view of the binary classification, the final FC was reset and the network was fine-tuned via transfer learning.

(3) Object detection deep learning network (Faster R-CNN, Cascade R-CNN, and SSD). In the object detection network of spikelet counting, each target can be detected and classified simultaneously. By classifying each spikelet in the image, we can obtain the number of *indica* spikelets and the number of *japonica* spikelets from one image. We assign a spikelet as *japonica* or *indica* if that type occupies the majority (more than 50%) of the image. The partition of dataset for these methods is the same as those used in spikelet counting.

### 2.10. Accuracy evaluation of three types of spikelet classification method

The rate of accuracy was used to evaluate the classification accuracy, using the following equation:

\[
\text{Accuracy} = \frac{N_{\text{true}}}{N_{\text{all}}}
\]  

(4)

where \(N_{\text{true}}\) is the number of test samples that are accurately predicted and \(N_{\text{all}}\) is the total number of test sets.

In the image classification method (SVM, image classification deep learning network) for the whole panicle, the category of the image can be directly assigned. However, because in the object detection method with the deep learning network (Faster R-CNN, Cascade R-CNN, SSD), the category of each object can be identified but the category of the image cannot be directly obtained, multiple indicators were used to evaluate the effect of the object detection deep learning model in the process of *indica* vs. *japonica* classification. First, the category with more than 50% of the objects in an image was assigned as the category of the image and equation 4 was applied to evaluate the accuracy of *indica* vs. *japonica* classification. Then mean average precision (mAP), recall, precision, and \(F\)-values were used to evaluate the quality of the deep learning model. The calculation equations were as follows:

\[
\text{Recall} = \frac{TP}{TP + FN}
\]  

(5)

\[
\text{Precision} = \frac{TP}{TP + FP}
\]  

(6)

\[
F = \frac{2TP}{2TP + FP + FN}
\]  

(7)

where TP is true positive, FP is false positive, FN is false negative.

### 3. Results and discussion

#### 3.1. Comparison of spikelet counting results with 5 methods

From the results (Fig. 7 and Table 3), the prediction accuracy of the two-stage deep learning method (Faster R-CNN and Cascade R-CNN) are better than those of the other three methods (Area, P-TRAP, and SSD). Cascade R-CNN reaches an \(R^2\) of 0.99 and MAPE is less than 1.42%, while Faster R-CNN attains an \(R^2\) of 0.99 and MAPE
is less than 1.68%. The measurement accuracy of the two-stage model (Faster R-CNN and Cascade R-CNN) is much higher than that of the one-stage model (SSD), indicating that the two-stage model showed better performance in the detection of small targets. In terms of time consumption, the one-stage model was much faster than the two-stage model, and Faster R-CNN was slightly faster than Cascade R-CNN (Table 3).

Table 3 – Measurement accuracy and training time consumption of the spikelet number using 5 methods

<table>
<thead>
<tr>
<th>Method</th>
<th>$R^2$</th>
<th>MAPE (%)</th>
<th>RMSE</th>
<th>Inference time per image (s)</th>
<th>Training time per epoch (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area method</td>
<td>0.80</td>
<td>29.04</td>
<td>70.73</td>
<td>1.23</td>
<td>-</td>
</tr>
<tr>
<td>P-TRAP</td>
<td>0.48</td>
<td>8.13</td>
<td>24.01</td>
<td>40.00</td>
<td>-</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>0.99</td>
<td>1.42</td>
<td>4.36</td>
<td>0.10</td>
<td>42.00</td>
</tr>
<tr>
<td>Cascade R-CNN</td>
<td>0.99</td>
<td>1.63</td>
<td>4.94</td>
<td>0.15</td>
<td>60.00</td>
</tr>
<tr>
<td>SSD</td>
<td>0.10</td>
<td>29.73</td>
<td>69.56</td>
<td>0.34</td>
<td>100.00</td>
</tr>
</tbody>
</table>

P-TRAP, a Panicle Trait Phenotyping tool; R-CNN, region-based convolutional neural network; SSD, Single Shot MultiBox Detector; MAPE, mean absolute percentage error; RMSE, root mean square error. - , no training required.

Compared with two-stage models, the prediction accuracy of the area method (Fig. 7a) decreased because some adhering spikelets led to the inaccurate calculation of TSA. In the P-TRAP method (Fig. 7b), there was some misjudgment using P-TRAP (shown in Fig. 8c–e). For example, the main stem (Fig. 8e) and some noise (Fig. 8c) were misassigned as spikelets, and some spikelets could not be successfully identified (Fig. 8d). In contrast, Faster R-CNN (Fig. 7c) did not experience these problems (Fig. 8b and f). In fact, the accuracy of quantitative detection of P-TRAP relies on the quality of the images. If impurities (worms, stains, etc.) look similar to spikelets, P-TRAP will require more time for manual correction. The prediction accuracy of SSD decreased (Fig. 7e) mainly because the resized image resolution of SSD (512 × 512 pixels) was much lower than those of two-stage models (1333 × 800 pixels).
Fig. 7 – Prediction results for spikelet number by five methods.
Regression scatter diagram of prediction results for spikelets counting using (a) area method, (b) P-TRAP, (c) Faster R-CNN, (d) Cascade R-CNN, (e) SSD.
P-TRAP, a Panicle Trait Phenotyping tool; R-CNN, region-based convolutional neural network; SSD, Single Shot MultiBox Detector.

Fig. 8 – Comparison of spikelet identification using P-TRAP and Faster R-CNN.
(a) Examples of prediction results for P-TRAP. (b) Examples of prediction results for Faster R-CNN. Misassignment of P-TRAP for (c) background, (d) spikelets, and (e) main stem. (f) Accurate assignment at the same position of Faster R-CNN.
P-TRAP, a Panicle Trait Phenotyping tool; R-CNN, region-based convolutional neural network.
We recommend Faster R-CNN to complete spikelet counting. Using the well-trained Faster R-CNN model, there is no need for any preprocessing, such as image cutting or denoising, and end-to-end prediction can be directly achieved. The role of hyperparameters is also crucial. One of the most important parameters is the number of RPN proposal boxes sent to the RoiAlign layer. The default value of the number of RPN proposal boxes in the Faster R-CNN model is 2000, and we found that using the default value did not achieve a good effect. The reason was that when one panicle has a large number of spikelets, the default value cannot generate enough proposals to detect all objects. When this value was increased to 3000, the proposals could cover more foregrounds and backgrounds, greatly improving the prediction effect. Also, owing to the large numbers of objects, as many as 200-300 per image, we need only a small number of images for training to achieve a good effect, greatly reducing the time cost of preparing the datasets.

3.2. Comparison of the classification results of indica and japonica with SVM, image classification, and object detection deep learning network

In the SVM method, we divided the traits involved in prediction into three groups: RGB traits, X-ray traits, and all traits. Table 4 shows the classification accuracy results. Figure 9 shows the P-R curves and F values of each object detection model, and Table 5 summarizes the mAP, recall, precision, and AP values of each class. Comparison of classification results including SVM using RGB traits, X-ray traits, and all traits indicate that the SVM algorithm using 45 RGB and X-ray traits attained the highest accuracy rate of 0.88. Using the recursive feature elimination (RFE) method, several traits with the highest trait contribution were also found, namely primary axis length (PL), SD of spikelet circularity (SD_SC), seed setting rate (SSR), spikelet circularity (SC), and SD of spikelet area (SD_SA).

<table>
<thead>
<tr>
<th>Method types</th>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image classification</td>
<td>Image classification (Res50)</td>
<td>0.96</td>
</tr>
<tr>
<td>Object detection</td>
<td>Faster R-CNN</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>Cascade R-CNN</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>SSD</td>
<td>0.76</td>
</tr>
<tr>
<td>SVM</td>
<td>SVM (RGB)</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>SVM (X-ray)</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>SVM (All)</td>
<td>0.88</td>
</tr>
</tbody>
</table>

R-CNN, region-based convolutional neural network; SSD, Single Shot MultiBox Detector; SVM, support vector machine.
Table 4 shows that the deep learning method attained the highest accuracy of accession classification, and the classification effect of the full image was best, reaching 0.96. The classification effect of Faster R-CNN was greater than that of Cascade R-CNN. The effect of SSD was the worst. In the object detection model, the detection and classification effects of the two-stage model were much better than those of the one-stage model (Table 5).

The results reveal that deep learning is an efficient method to realize the classification of indica and japonica accessions. Although the SVM algorithm can also achieve a high accuracy rate, it needs two types (X-ray and RGB images) of traits to be trained and implemented. In contrast, the deep learning method needs only RGB images to achieve high accuracy. The classification effect of the full image is better than that of the object detection model, but the object detection model can accurately classify and detect each spikelet simultaneously.
We conclude that Faster R-CNN attains a better effect than other methods in counting and classification of rice panicles.

3.3. Description and correlations of extracted digital traits

We obtained 46 traits, including spikelet, kernel, panicle structure, and comprehensive traits. Using RGB images, we obtained the number and morphological traits of spikelets, including length, circumference, and roundness, as well as the panicle structural traits of rice panicles. The definitions of the spikelet and kernel shape and panicle structure traits are shown in Fig. 10. Using X-ray images, we can directly obtain the shape traits of kernels without hulling and threshing (Fig. 10d). From these traits, more comprehensive traits can also be calculated, such as the seed setting rate (SSR) and kernel plumpness (KP) per panicle.

Fig. 10 – Examples of spikelet shape, kernel shape, and panicle structure traits.
(a) RGB image. (b) Spikelet in enlarged image. (c) X-ray image. (d) Kernel in enlarged image. (e) Panicle traits.

We next analyzed 200 samples of 40 accessions to calculate a total of 46 traits and used this batch of traits to draw a heatmap (Fig. 11) to illustrate the correlations between the 46 traits. Spikelet number was positively correlated ($r = 0.91$) with the number of secondary branches and ($r = 0.56$) with the number of primary branches.
The interval length of the secondary branch was negatively correlated \((r = -0.35)\) with the number of spikelets, implying that more denser branches can contribute more spikelets. Seed setting rate (SSR) was positively correlated with kernel number \((r = 0.65)\). The seed plumpness (SP) was positively correlated with kernel length \((r = 0.71)\), kernel width \((r = 0.66)\), kernel area \((r = 0.73)\), and kernel perimeter \((r = 0.75)\), indicating that the larger the kernel, the higher is the SP of the corresponding rice accession.

![Heatmap of pairwise trait correlations of 46 traits.](image)

Fig. 11 – Heatmap of pairwise trait correlations of 46 traits. Pearson correlation coefficient is used to show the correlation between all traits.

### 3.4. Evaluation of spikelet identification using Faster R-CNN under different scenarios

To test the robustness of spikelet identification using Faster R-CNN, we tested 10 panicles under 4 different scenarios: panicle without and with manual shaping under natural light illumination (Fig. 12a and e), panicle without and with manual shaping under backlight illumination (Fig. 12c and g). The results showed that the spikelet counting accuracy of panicles with manual shaping was higher than that of panicles without manual shaping, and the spikelet counting accuracy under backlight illumination was higher than that under natural light.
illumination. The $R^2$ of panicles without manual shaping under backlight illumination was as high as 0.76, indicating the potential of future application of the Faster R-CNN model in the field. We also tested the Faster R-CNN model to count the spikelets after threshing (Figs. 12i and k). The $R^2$ for spikelet counting under natural light illumination or backlight illumination were both greater than 0.97, indicating the potential of using Faster R-CNN for seed counting after threshing.

Fig. 12 – Evaluation of spikelet identification using Faster R-CNN under several scenarios.
(a, b) Panicle without manual shaping under natural light illumination. (c, d) Panicle without manual shaping under backlight illumination. (e, f) Panicle with manual shaping under natural light illumination. (g, h) Panicle with manual shaping under backlight illumination. (i, j) Threshed spikelets under natural light illumination. (k, l) Threshed spikelet under backlight illumination.

3.5. Integrated panicle phenotyping method benefits crop breeding

We developed a bi-model rice panicle phenotyping system and a corresponding image analysis pipeline to evaluate spikelet and kernel traits, with the following advantages:

(1) Conventional phenotyping of rice yield-related traits usually requires threshing and hulling, which are time-consuming. We have described an integrated panicle phenotyping method using RGB and X-ray images and an image analysis pipeline to evaluate 46 digital traits. This approach can improve measurement efficiency and provide new information for rice functional genomics and rice breeding.
We compared five methods for counting spikelets and found that Faster R-CNN was able to count spikelets with high accuracy ($R^2$ of 0.99) and speed. We also tested spikelet identification using Faster R-CNN under four different scenarios. The $R^2$ of panicle without manual shaping under backlight illumination (Fig. 12a) was as high as 0.76, suggesting that with a mobile backlight imaging device in the field and with more training samples, the Faster R-CNN model could potentially be extended to spikelet detection and counting in the field.

Using Faster R-CNN and RGB images of panicles, indica and japonica accessions could be accurately classified (with an accuracy of 0.91). If more rice accessions were included in the training set, the model could be extended to accession identification.

In this work, we focused on extracting the 2D traits of panicle, spikelet, and kernels. There are some 3D imaging techniques, for example X-ray Micro-CT, that have been used for measuring 3D grain traits of rice [43] or wheat [44]. Combining our method with X-ray CT, more kernel information, such as volume or surface area, could be extracted. However, to extract complex traits such as 3D panicle structure traits, a specific 3D image analysis pipeline for CT images should be further developed.

4. Conclusions

In this work, based on X-ray and RGB scanning, we developed an integrated rice panicle phenotyping system and corresponding image analysis pipeline to extract spikelet and kernel traits with high accuracy and efficiency, which will benefit rice functional genetics and rice breeding in the future.

Declaration of competing interest

Authors declare that there are no conflicts of interest.

Author contributions

Lejun Yu and Jiawei Shi developed the process programs, analyzed the data, and drafted the manuscript. Chenglong Huang, Lingfeng Duan, and Di Wu assisted in designing the phenotyping platform and Debao Fu helped to acquire the images. Debao Fu, Changyin Wu, and Lizhong Xiong contributed rice materials and rice management. Wanneng Yang and Qian Liu supervised the project and wrote the manuscript. All of the authors read and approved the final manuscript.
Acknowledgments

This work was supported by the National Key Research and Development Program of China (2016YFD0100101-18), the National Natural Science Foundation of China (31770397, 31701317), and the Fundamental Research Funds for the Central Universities (2662017PY058).

Data availability

Supplementary data for this article can be accessed via

References


