Research Article

Received: 24 December 2021 Revised: 18 February 2022 Accepted article published: 2 March 2022 Published online in Wiley Online Library:

(wileyonlinelibrary.com) DOI 10.1002/jsfa.11848

Fast location and segmentation of high-throughput damaged soybean seeds with invertible neural networks

Ziliang Huang,[a,b] Rujing Wang,[a] Qiong Zhou,[a,b] Yue Teng,[a,b] Shijian Zheng,[c] Liu Liu[d*] and Liusan Wang[a*]

Abstract

BACKGROUND: Fast identification of damaged soybean seeds has undeniable importance in seed sorting and food quality. Mechanical vibration is generally used in soybean seed sorting, but this can seriously damage soybean seeds. The convolutional neural network (CNN) is considered an effective method for location and segmentation tasks. However, a CNN requires a large amount of ground truth data and has high computational cost.

RESULTS: First, we propose a self-supervision manner to automatically generate ground truths, which can theoretically create an almost unlimited number of labeled images. Second, instead of using popular CNNs, a novel invertible convolution (involution)-enabled scheme is proposed by using the bottleneck block of the residual networks. Third, a feature selection feature pyramid network (FS-FPN) based on involution is designed, which selects features more flexibly and adaptively. We further merge involution-based backbones and FS-FPN into a unified network, achieving an end-to-end seed location and segmentation model; the best mean average precision of location and segmentation achieved was 85.1% and 81% respectively.

CONCLUSION: The experimental results demonstrate that the proposed method greatly improves the performance of the baseline network with faster speed and fewer parameters, enabling it to detect soybean seeds more effectively.

Keywords: high-throughput soybean seeds; location; segmentation; invertible neural networks; feature pyramid network

INTRODUCTION

The soybean is one of the most important leguminous species, ranking among the top five major crops in the world. It is widely consumed for its high protein and vegetable oil content.[1] Crop yield and consumption of soybean depend highly on the quality of soybean seeds. The quality is strongly associated with the economic value, which is of great significance to farmers and even the country. To reach a high standard of seeds quality, it is necessary to develop fast and accurate methods of detecting the appearance quality of soybean seeds. The most widely used grading machine uses mechanical vibration to select the soybean seeds. This can effectively separate large-sized debris and seeds with non-standard shapes. However, the vibration method can cause serious damage through violent collisions; and it cannot detect low-grade seeds, such as insect-bored seeds, spotted seeds, and heterochromatic seeds with a similar shape to perfect soybean seeds.

The development of machine vision has led to it becoming widespread in the field of agriculture,[2,3] such as in identifying the number of aflatoxin-contaminated pistachio and cashew nuts,[4] objective and accurate identification of tea grading,[5] discriminating pepper seeds,[6] and classifying wheat grains.[7] Momin *et al.*[8] developed a method in the detection of materials other than grain in soybean harvesting. Tan *et al.*[9] used traditional computer vision technology with back-propagation neural networks to classify four types of damaged soybean seeds. Liu *et al.*[10] analyzed the image characteristics of different damaged soybean kernels. Although traditional image-based machine vision studies are modified to solve the challenges of detecting soybean seeds, there are still some critical problems that hinder its development. First, shadow noise and inconsistent illumination conditions can occur during capturing seed images. Second, color difference and shape difference can degrade the accuracy of detection. Third, densely sampled seeds cannot be effectively separated into different distributions. The classic manual-designed machine vision methods are sensitive to the noise, illumination, and texture of objects, which show weak robustness and poor generalization ability.

* *Correspondence to: L Liu, Shanghai JiaoTong University, Shanghai 200240, China, E-mail: liuliu1993@sjtu.edu.cn; or L Wang, Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Science, Hefei 230031, China, E-mail: lswang@iim.ac.cn*

a *Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Science, Hefei, China*

b *University of Science and Technology of China, Hefei, China*

c *Southwest University of Science and Technology, Mianyang, China*

d *Shanghai JiaoTong University, Shanghai, China*

Recently, convolutional neural networks (CNNs) have become dominant in the field of deep learning, including object detection,[11,12] object segmentation,[13,14] and classification.[15] CNNs have also been used in agriculture research, Mubin *et al.*[16] predicted and counted oil palms in satellite imagery by utilizing two different CNNs. Wang *et al.*[17] proposed a multi-projection pest detection model to overcome the imbalance of classes. Li *et al.*[18] designed a coarse-to-fine network for aphid detection in dense distribution regions. Veeramani *et al.*[19] used a novel application for a deep convolutional network to sort haploid maize seeds. The study outperformed existing state-of-the-art traditional machine vision classifiers. Steinbrener *et al.*[20] showed an easy way to classify hyperspectral images with an RGB pre-trained network. Bosilj *et al.*[21] explored the effectiveness of knowledge transfer between CNN-based classifiers for different crop types and obtained a reduction in training times of up to 80%. Misra *et al.*[22] proposed the use of SpikeSegNet to classify and count the spikes of wheat plants, and the testing performance demonstrated the proposed approach was effective and robust. CNN-based algorithms can improve recognition accuracy and greatly simplify the complex process of feature extraction. However, training a deep-learning model with strong robustness and generalization ability requires a huge amount of labeled data. In addition, most CNNs rely on deep layers and complex computation, and such ponderousness undoubtedly makes the existing models less practical for resource-constrained applications.

Involution has symmetrically inverse inherent properties compared with standard convolution. It can promote the efficiency of visual recognition in a lightweight manner by embedding switchable and scalable modeling into the representation learning paradigm. Invertible CNNs (INNs) are characterized by the mapping from inputs to outputs being bijective, and both mappings have a tractable Jacobian. Meanwhile, the mapping of forward and inverse are efficiently computable.[23,24] Concisely speaking, INNs are very suitable for seed recognition tasks owing to the low complexity in the spatial and channel domains.

In this paper, inspired by successful applications in Varol *et al.*[25] and Ward *et al.*,[26] we first propose to construct a primitive repository that is composited from single soybean seeds and rebuild the primitives to generate sufficient training data with automatically generated ground-truth annotations. In this way, our model can be trained in a self-supervision manner on the composed dataset, and the primitive-trained model can be easily transferred to real-world data. Second, to achieve the purpose of fast location and segmentation of the high-throughput soybean seeds, we leverage the inherent advantages of Mask R-CNN[27] and involution, proposing an involution-based Mask R-CNN; we term this model with involution backbone as Inv-Mask. Third, we design the feature selection (FS) feature pyramid network (FPN) based on involution to tackle the serious information loss in a standard FPN. The FS-FPN ensures the effectiveness in extracting multiscale features, generating more valuable features and enhancing semantic consistency.

In summary, our primary contributions are as follows:

- We propose a self-supervised method of constructing sufficient training data with automatically generated ground-truth annotations. This can significantly decrease the labor cost for data creation and conquer the scarcity of data in training a task-specified agricultural model.
- We build a hybrid synthetic/real dataset of damaged soybean seeds. We train and evaluate our model from simulation data to real-world data. This is the first attempt to utilize a synthetic dataset for the detection of high-throughput damaged soybean seeds.
- We present the Inv-Mask model, which is an involution-based efficient network with faster speed and higher performance. We first demonstrate the involution-powered deep-learning model works well in the field of seeds location and segmentation.
- We design a novel FS-FPN based on involution operation to aggregate more discriminative features through attention-guided FS. FS-FPN not only leverages the inherent feature hierarchy, fusing adjacent features through lateral connections and a top-down pathway but also relieves the information loss in dimension reduction.

MATERIALS AND METHODS

Dataset

Synthetic dataset creation

We randomly chose 700 soybean seeds, including broken kernels, heterochromatic kernels, insect-bored kernels, moldy kernels, perfect kernels, split kernels, and spotted kernels. We place every single seed above a platform with a black background. The seed images were captured using an MV-CE200 industrial camera (Hikvision Digital Technology Co. Ltd, Hangzhou, China) and saved as individual image files.

We used a self-supervision manner to build the training data for learning primitive knowledge. Specifically, we prepared 1000 black background images as the background image set (BIS) with a fixed size of 1024×1024 px^2. Then, for the 700 single soybean seed images, we removed the background regions and prepared a seed image set (SIS). The BIS and SIS were used to create a synthetic image dataset in a self-supervised manner, the process is described as follows. The initial background of SIS is black, we used a classic threshold segmentation algorithm to subtract the background and get regions of interest, such as the seed area, which is small compared with the entire image. Second, an image was randomly chosen from the BIS and pasted to a raw image canvas with the same size. Third, we randomly selected a seed image from the SIS, and the rotation angle was randomly set. After getting the processed seed image, we paste it on the coordinate (x,y) of the raw image canvas. The coordinate was randomly determined but restricted to a certain algorithm, namely constrained domain randomization, as shown in Eqn (1). This formula was used to make sure that the image would not exceed the canvas size and control the distance of each seed. In addition, Eqn (1) ensures the overlapping proportion of the seed pasted area. If the distance or the overlapping value cannot meet the minimum threshold, pasting is exited and restarted to choose a new coordinate. During the pasting process, generate the corresponding mask by creating a black background canvas and coloring the seed area with a specific color. This operation automatically annotates each seed using different colors. Finally, the operation was repeated until the coordinate cannot find a position that meets our requirements. The procedure of creating the synthetic image dataset is shown in Fig. 1.

$$\mathrm{coord}(x_i,y_i)=\begin{cases} C_x-B_w/2\le x_i\le C_x+B_w/2 \\ C_y-B_h/2\le y_i\le C_y+B_h/2 \quad i=1 \\ \mathrm{sign}^*(x_i,y_i)\sqrt{|x_i-x_j||y_i-y_j|}\ge T(I,J) \quad i>1, 1\le j\le(i-1) \end{cases} \tag{1}$$

$$\mathrm{sign}^*(x_i,y_i)=\begin{cases} 1 & (0,0)\le(x_i,y_i)\le(B_h-I_h,B_w-I_w) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$T(I,J)=\beta_d\mathrm{Diagonal}(I,J) \tag{3}$$

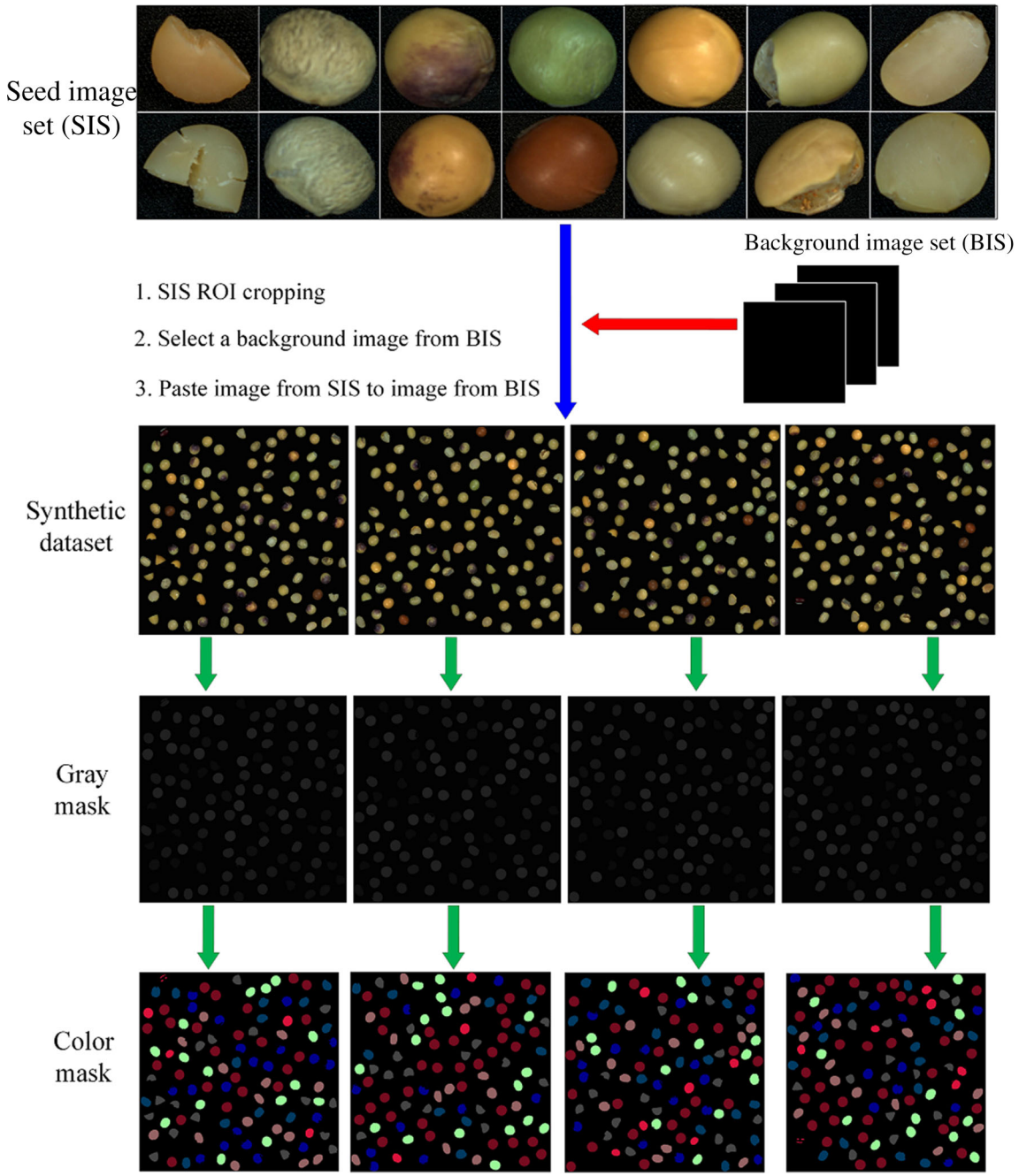

Figure 1. Synthetic dataset generation process. The gray mask and color mask will be generated while generating synthetic images. ROI: region of interest.

where coord(x_i,y_i) is the coordinate of the ith seed image pasted on the canvas, B_h and B_w represent the height and width of the canvas respectively, C_x and C_y are the center points of the input image, sign$^*(x_i,y_i)$ is a flag function indicating whether the coordinate will be allowed in the operation, and $T(I,J)$ is the threshold distance and determines whether the seed image is saved or not. In Eqn (2), I_h and I_w represent the height and width of the seed image respectively. The pasted seed image should be inside the canvas; if not, we discard the coordinate. $T(I,J)$ depends on the diagonal distance of the ith to jth (1st $\sim i-$ 1st) seed image. We also propose a hyperparameter β_d to control the density of pasting data, as shown in Eqn (3).

Real-world dataset creation

After we created the synthetic dataset using our the generation algorithm, we also prepared a real damaged soybean seed dataset with image size 1024 × 1024 px^2. We randomly put soybean seeds on a conveyor and shot them during transmission. In the acquisition process, to simulate the uncontrollable lighting conditions in the real environment, we used two adjustable light sources to enhance the authenticity of the dataset. Then, we removed seed images that were incomplete or at the edges from all images, and 500 images were finally retained (as shown in Fig. 2). The real images were labeled using the professional annotation software LabelMe.[28]

Model details

Architecture of our method

Mask R-CNN[27] follows the philosophy of Faster R-CNN[11] and extends a new branch for the instance segmentation tasks. In this way, the segmentation branch parallels with the branch of classification and detection in Faster R-CNN, which is an extremely efficient design to generalize to other tasks. In general, Mask R-CNN consists of four parts: (i) an encoder network model to extract features, namely backbone; (ii) an FPN to aggregate features at different scales; (iii) a region proposal network to generate detection bounding boxes; and (iv) three heads, namely a classification head, a detection head, and a segmentation head. Our Inv-Mask model is designed on the basis of Mask R-CNN, which employs the backbone based on involution. We show the structure of Inv-Mask + FS-FPN in Fig. 3; we set the backbone structure the same as ResNet, and the building block shows in Fig. 4. To be specific, we replace the 3×3 convolution layer in the original ResNet with involution, and we retain the 1×1 convolution for channel projection and fusion.

Invertible neural networks

INNs are a type of bijective neural network mapping from inputs to outputs; these not only predict the output given the input, but also predict the input given an output. The special characteristic of INNs means they can learn the forward and inverse mappings at the same time. INNs can also adaptively assign weights to different positions, thereby giving priority to the visual elements with the largest amount of information in the spatial domain. Various invertible neural networks have been designed: Maclaurin *et al.*[29] proposed that the hyperparameter gradient is calculated by using momentum to accurately reverse the dynamics of stochastic gradient descent; and Dinh *et al.*[30] explored a new flexible architecture for learning highly nonlinear bijective transformations, mapping training data to a factorized distribution space, and later[31] they presented a class of invertible functions with tractable Jacobian determinants, which performed well in generative models. Inspired by the former studies, we show the working principle of invertible neural networks in the following. INNs must partition the units into two groups in each layer, which we denote as X_1 and X_2. Figure 5 shows the architecture of the invertible block. Each invertible block has inputs (X_1, X_2) and the corresponding outputs (Y_1, Y_2). We demonstrate this as follows:

$$\begin{aligned} Y_1 &= X_1 \odot \exp(\mathcal{F}(X_2)) + \mathcal{G}(X_2) \\ Y_2 &= X_2 \odot \exp(\mathcal{F}(Y_1)) + \mathcal{G}(Y_1) \end{aligned} \tag{4}$$

The functions $\mathcal{F}$ and $\mathcal{G}$ are analogous to those in standard ResNet,[32] and $\odot$ represents the elementwise or Hadamard product. Each layer's activations can be easily inverted from the next layer's activations:

$$\begin{aligned} X_2 &= (Y_2 - \mathcal{G}(Y_1)) \odot \exp(-\mathcal{F}(Y_1)) \\ X_1 &= (Y_1 - \mathcal{G}(X_2)) \odot \exp(-\mathcal{F}(X_2)) \end{aligned} \tag{5}$$

A deep invertible neural network is made up by stacking these reversible blocks. In the standard architecture, it is assumed the inputs and outputs

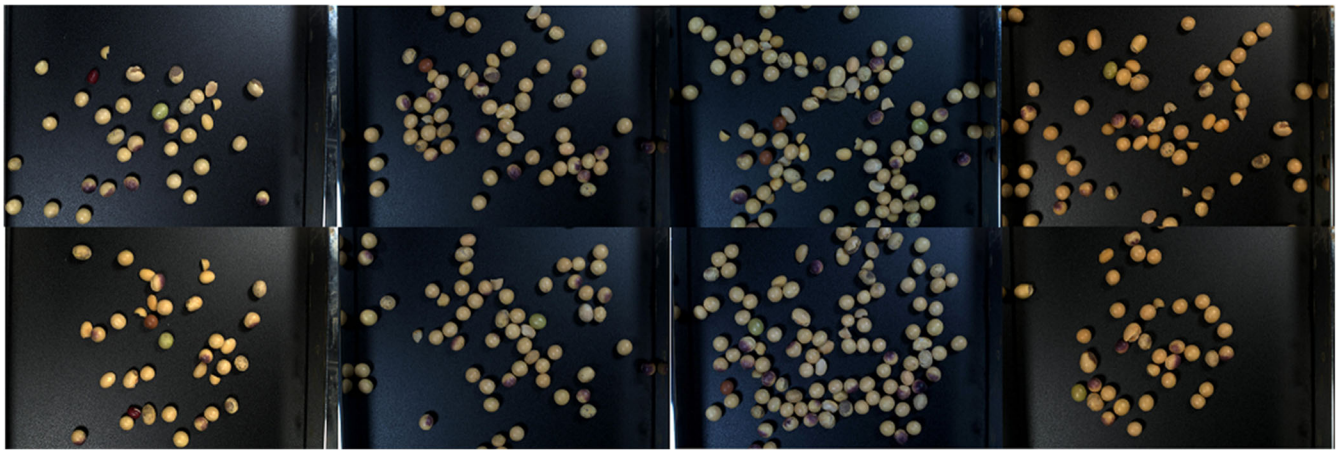

Figure 2. Real-world images under unstructured illumination.

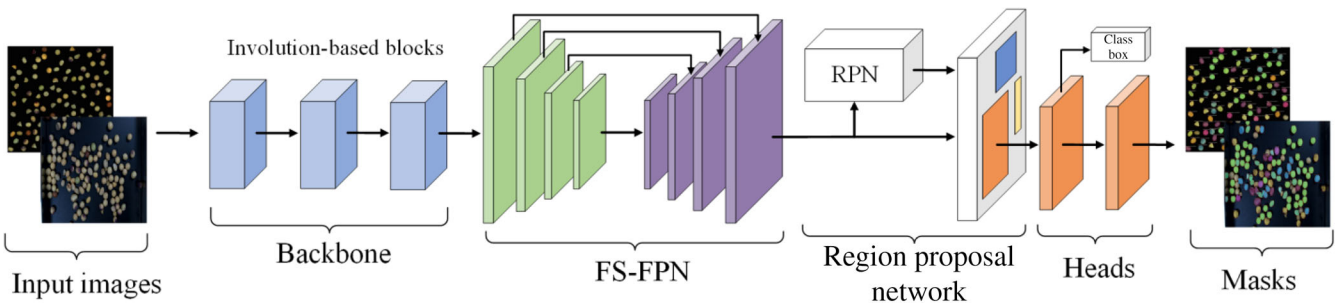

Figure 3. Structure of the proposed Inv-Mask + FS-FPN. FS-FPN: feature selection feature pyramid network; RPN: region proposal network.

have the same dimension. If not, we pad the inputs and outputs with an equal number of zeros if Dimension(inputs)<Dimension(outputs). This enables the interior layers of the network to learn a complex transformation in a more flexible way and embed features into a larger presentation space. For designing the experimental network with involution, we imitate the design of ResNet owing to its excellent architecture. We stack residual blocks as ResNet but replace the 3×3 convolution with 3×3 involution at all bottlenecks. Considering the channel fusion and projection, we retain the initial setup of ResNet.

FS-FPN

To remedy the issue that occurred in standard FPN, we propose a novel FS-FPN, as shown in Fig. 6. The details are described in the following.

Channel-feature selection block. Feature extraction in FPN usually uses the lateral connections with 1 × 1 convolution to generate the same dimension features. The method is simple but suffers from inevitable information loss due to the direct channel dimension reduction. As shown in Fig. 6, we propose a channel selection block (CSB) to emphasize the important feature maps and locate objects more accurately based on the attention mechanism. The details of the CSB in Fig. 6 are presented in the following. $Z=[Z_1,Z_2,\ldots,Z_N]$ and $M=[M_1,M_2,\ldots,M_{N'}]$ present the input and output FPN layers respectively. $Z_i=[z^1,z^2,\ldots,z^D]$ and $M_i=[m^1,m^2,\ldots,m^{D'}]$, where D and D' represent the number of feature maps in Z_i and M_i respectively. $M_i=[m^1,m^2,\ldots,m^{D'}]$ is the result after involution and F_{cs}; the purpose is to find the most valuable feature maps and drop the redundant features. The process of CSB is presented as follows:

$$\begin{aligned}\Phi\left(z_{i,j}^{d}\right)&=W_1\sigma\left(W_0 z_{i,j}^{d}\right)\\ m_{i,j}^{d'}&=F_{cs}\left(\Phi\left(z_{i,j}^{d}\right)\right)\end{aligned}\tag{6}$$

where i,j is the coordinate in the dth feature map. $W_1\in\mathbb{R}^{(N\times N\times G)\times(C/r)}$ and $W_0\in\mathbb{R}^{(C/r)\times C}$ stand for two linear transformations, where G represents the group number of involution and r is the reduction ratio.

A CSB is utilized in the process of multiscale feature aggregation. Different from the general attention mechanism module, the module we designed is embedded in FPN, reducing the information loss during horizontal connection. At the same time, the CSB could adaptively recalibrate the importance of each channel, which is inspired by the squeeze-and-excitation network.[33] CSB commits to the meaningful channel of an input image.

Spatial-feature attention block. Since each seed belongs to a key region in the image, to improve the feature expression of the key region, we introduce a spatial-feature selection block (SSB) to utilize the inter-spatial relationship of features. We adopt two operations, global max pooling and global average pooling, generating two feature descriptors representing different information. Then, the two feature maps are fused by using the receptive field with the size of $k\times k$, and we apply involution layers to encode the feature map where to emphasize or suppress. SSB can be expressed as

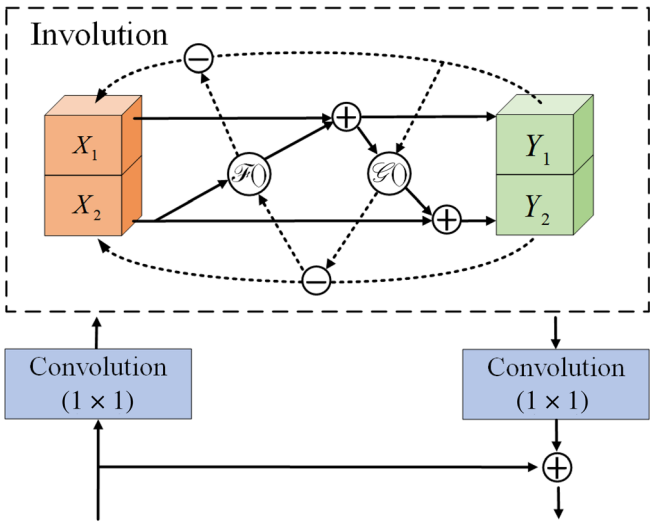

Figure 4. Building block of the Inv-Mask backbone. The solid line represents forward pass, and the dotted line indicates inverse pass.

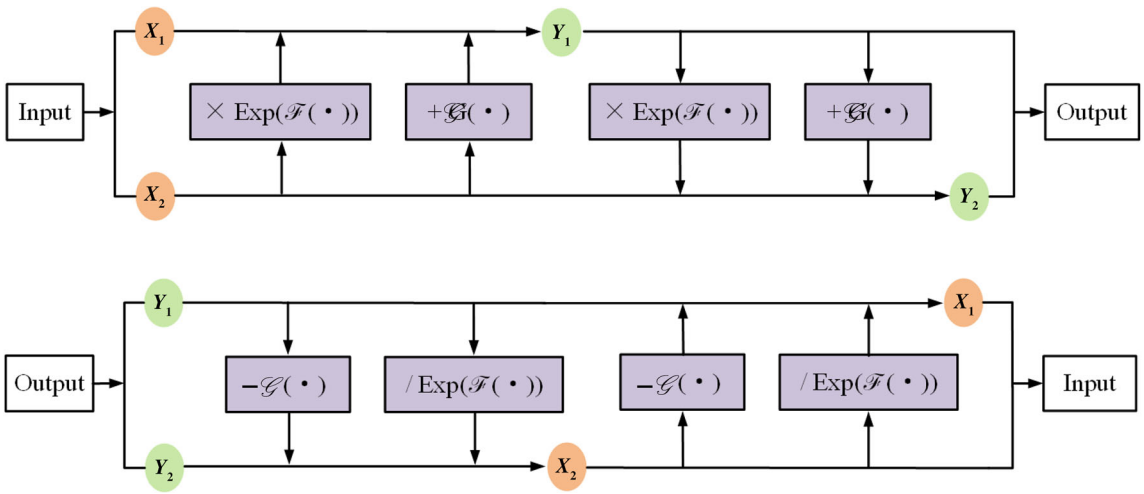

Figure 5. The forward pass (up) and inverse pass (down) of invertible neural networks.

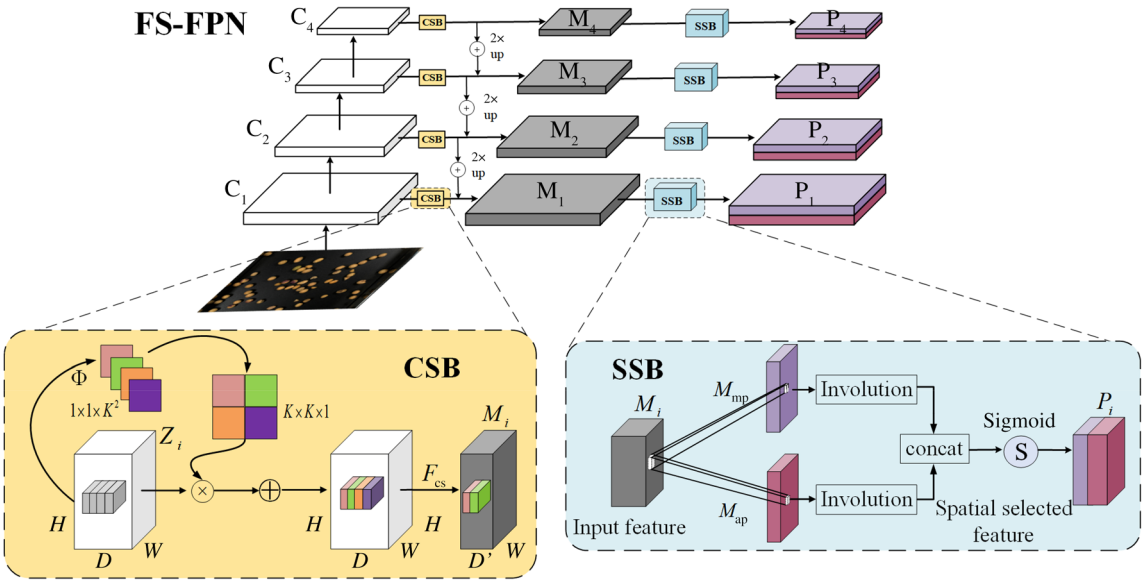

Figure 6. The architecture of the feature selection feature pyramid network (FS-FPN). We display the details of channel selection block (CSB) and spatial-feature selection block (SSB) in the dashed boxes below.

$$P^{d'} = \sigma\left(\text{concat}\left(\left[\text{Inv}\left(M_{\text{mp}}^{d'}\right), \text{Inv}\left(M_{\text{ap}}^{d'}\right)\right]\right)\right) \tag{7}$$

where $P_i = [p^1, p^2, \ldots, p^{D'}]$, $M_{\text{mp}}^{d'}$ and $M_{\text{ap}}^{d'}$ denote the feature maps operated by maximum pooling and average pooling respectively, and σ is the sigmoid function. In essence, the spatial information in the original image is transformed into another space through the SSB, and the key information is preserved. Then, a weighted mask is generated for each position and weighted output so as to enhance the specific target area of interest and weaken the irrelevant background area.

EXPERIMENTS AND DISCUSSION

Experiment details

Experimental setup

In our experiments, we employed a stochastic gradient descent optimizer in the training process and set the learning rate to 0.0025 in the initial state, training for a total of 12 epochs. The weight decay constant was 1×10^{-3}, and the momentum was 0.9. The experimental analysis and data processing were performed using the Pytorch deep-learning architecture on a machine with an Intel i9-9900 k CPU, 128GB RAM and one piece of NVIDIA TITAN RTX GPU. Following standard segmentation detectors' training strategy, the networks we proposed were pre-trained on the synthetic dataset at first and then trained on the real-world dataset. All the models were trained by the way of end-to-end.

Evaluation metrics

Different measures were adopted in object detection and segmentation tasks. In general, true positive (TP), false positive (FP), true negative, and false negative (FN) are calculated to reveal the results of a model prediction. Precision and recall represent the performance from the perspective of prediction results and real results:

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}} \\ \text{recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \end{aligned} \tag{8}$$

Intersection over union (IoU) is a well-known indicator in detection and segmentation due to its strict criteria about over- and under-overlapping. It reflects the overlap ratio of two regions (usually the predicted area and the ground truth), as shown in Fig. **7**. IoU has two definitions, one being the bounding boxes IoU and the other being the masks IoU, and they can be calculated as follows:

$$\begin{aligned} \text{IoU} &= \frac{G_{\text{b}} \cap P_{\text{b}}}{G_{\text{b}} \cup P_{\text{b}}} \\ \text{MaskIoU} &= \frac{G_{\text{m}} \cap P_{\text{m}}}{G_{\text{m}} \cup P_{\text{m}}} \end{aligned} \tag{9}$$

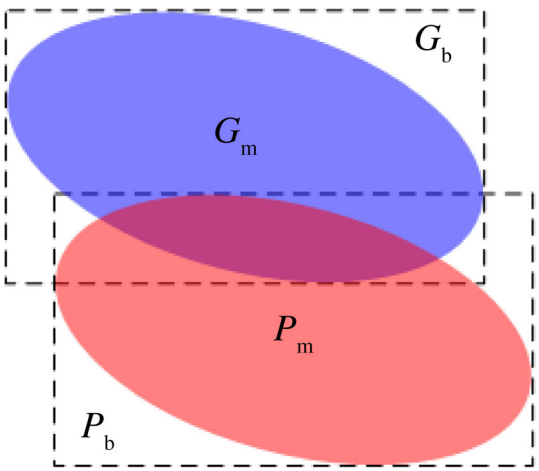

Figure 7. The intersection-over-union definitions of bounding boxes and masks. P_b represents the predict bounding box and G_b represents the ground-truth bounding box. P_m denotes the predict mask area and G_m denotes the ground-truth mask area.

Average precision (AP) measures the area under the curves of precision and recall with different thresholds. AP@50 (AP_{50}) is the prediction accuracy rate when the IoU = 0.5, AP@75 (AP_{75}) is the rate when the IoU = 0.75. AP@[0.5 : 0.95] indicates the threshold levels from 0.5 to 0.95 with step size of 0.05; namely, mean average precision (mAP).

Experiment results

As mentioned in the 'Real-world dataset creation' section, the synthetic dataset contains 1000 image pairs. We divide this into three parts: 800 pairs for training, 100 pairs for validation, and 100 pairs for testing. The real dataset contains 500 image pairs. We also split it into three parts: 350 pairs for training, 100 pairs for validation, and 50 pairs for testing. We report the evaluation values mentioned before as the standard Common Objects in Context (COCO) metrics, we use the bounding box AP (AP^{bb}) to show the accuracy of location and the mask segmentation AP (AP^{mk}) to show the accuracy of segmentation. According to the COCO evaluation criteria, AP_{50} is the most important criterion, and mAP is in second position.

Model performance on synthetic dataset

We first perform the models, including raw Mask R-CNN, our Inv-Mask, Inv-Mask + CSB, Inv-Mask + CSB + SSB (equivalent to Inv-Mask + FS-FPN), training from scratch on the synthetic dataset. The performance of the models is displayed in Table 1. With almost all groups in the table, the backbones of ResNet50 achieve a comparable AP to the backbones of ResNet101, so we focus on the models with the ResNet50 backbone. As we can see, Inv-Mask achieves a great margin of 8.5% higher bounding box AP_{50} and 8.4% higher mask AP_{50} over Mask R-CNN. The CSB and SSB employed in Inv-Mask further improve the performance in detection and segmentation. Our model with those two blocks could surpass the Mask R-CNN by 16.6%/16.6% AP_{50} and 19.9%/15% mAP in bounding box and mask respectively. Figure 8 shows the comparison of Mask R-CNN (a) and our model with the FS-FPN approach (b) on the heterochromatic category. The precision–recall curve takes recall as the abscissa and precision as the ordinate, and the performance of the model can be reflected by the area below the curve. The larger the area, the better the performance of the model. It can be seen intuitively from the picture, our model shows superior performance over Mask

Table 1. Location and segmentation results on synthetic dataset

Model	Backbone	AP^{bb}_{50}	AP^{bb}_{75}	mAP^{bb}	AP^{mk}_{50}	AP^{mk}_{75}	mAP^{mk}
Mask R-CNN	r50	0.772	0.756	0.692	0.772	0.757	0.590
Inv-Mask		0.857	0.847	0.757	0.856	0.837	0.725
Inv-Mask+CSB		0.932	0.918	0.886	0.932	0.908	0.826
Inv-Mask+CSB+SSB		**0.938**	**0.924**	**0.891**	**0.938**	**0.919**	**0.840**
Mask R-CNN	r101	0.711	0.692	0.641	0.711	0.708	0.645
Inv-Mask		0.842	0.834	0.796	0.795	0.775	0.703
Inv-Mask+CSB		**0.929**	**0.915**	0.887	**0.929**	**0.915**	0.845
Inv-Mask+CSB+SSB		0.925	0.915	**0.891**	0.925	0.904	**0.850**

r50 and r101 represent ResNet50 and ResNet101 respectively.
AP: average precision; bb: bounding box; mAP: mean average precision; mk: mask segmentation.

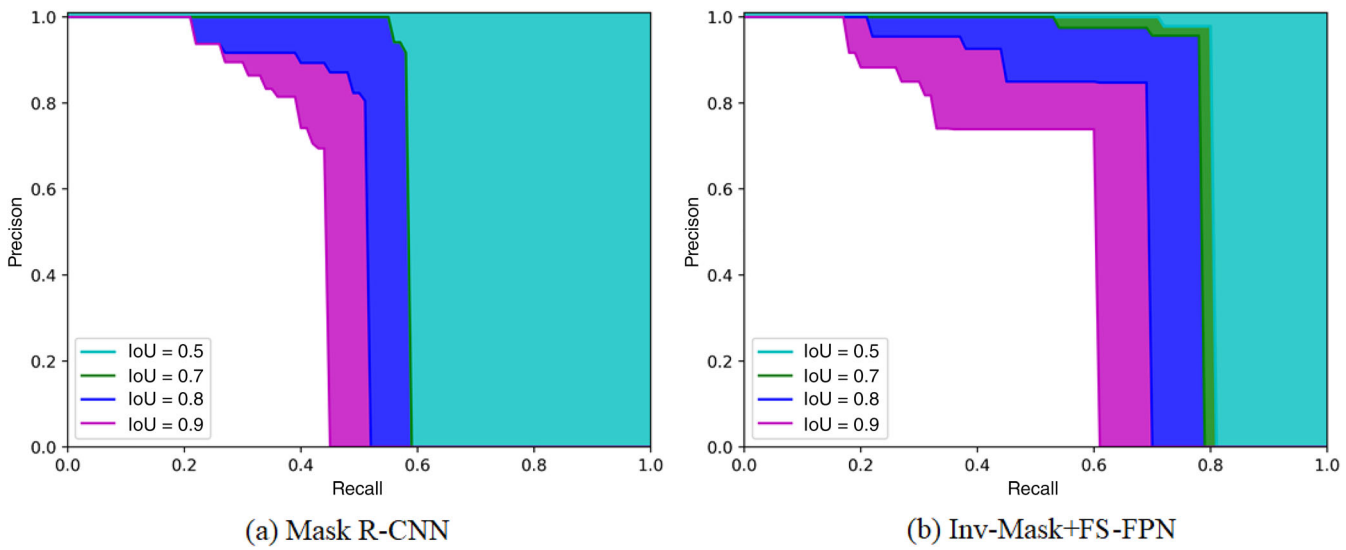

Figure 8. Quantitative evaluation of segmentation performance on the synthetic dataset. Performance comparison on heterochromatic category. FS-FPN: feature selection feature pyramid network; IoU: intersection over union.

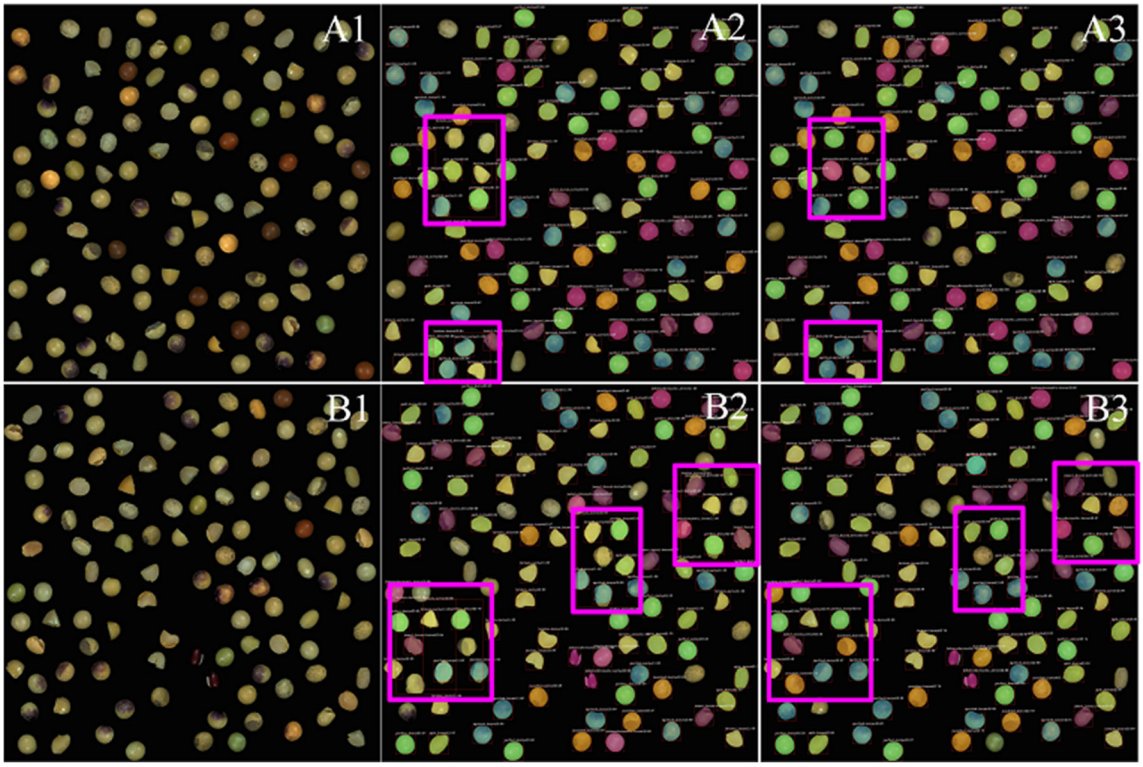

Figure 9. Visualized results of the synthetic dataset: (A1, B1) raw image; (A2, B2) the result of Mask R-CNN; (A3, B3) the result of Inv-Mask + FS-FPN. The pink boxes highlight the comparison at the same location. FS-FPN: feature selection feature pyramid network.

Table 2. Location and segmentation results on real-world dataset

Model	Backbone	AP^{bb}_{50}	AP^{bb}_{75}	mAP^{bb}	AP^{mk}_{50}	AP^{mk}_{75}	mAP^{mk}
Mask R-CNN	r50	0.725	0.601	0.542	0.725	0.656	0.546
Inv-Mask		0.842	0.802	0.671	0.842	0.800	0.675
Inv-Mask+CSB		0.892	0.886	0.842	0.892	0.879	0.795
Inv-Mask+CSB+SSB		**0.903**	**0.891**	**0.851**	**0.903**	**0.889**	**0.810**
Mask R-CNN	r101	0.722	0.623	0.547	0.721	0.654	0.551
Inv-Mask		0.842	0.793	0.668	0.842	0.812	0.673
Inv-Mask+CSB		0.889	0.885	**0.851**	0.889	0.885	**0.809**
Inv-Mask+CSB+SSB		**0.892**	**0.885**	0.847	**0.892**	**0.871**	0.805

r50 and r101 represent ResNet50 and ResNet101 respectively.
AP: average precision; bb: bounding box; mAP: mean average precision; mk: mask segmentation.

R-CNN. In terms of qualitative evaluation, we display some of the results in Fig. 9. The first column shows the raw images, the second column shows the Mask R-CNN results, and the third column shows the Inv-Mask + FS-FPN results. The pink boxes indicate that bounding boxes overlap occurs during detection in the Mask R-CNN, but our model performs well in the corresponding positions. It turns out that the model we proposed could help the popular Mask R-CNN improve performance on our synthetic dataset.

Model performance on real-world dataset

For the real-world application, we conduct experiments on our real dataset, and the pre-trained network backbones are those trained in the synthetic dataset. The results are shown in Table 2. As we can see, our proposed Inv-Mask pushes the envelope of precision boundary to a new level. The backbones of ResNet101 show a similar performance with the backbones of ResNet50, therefore, we will report the ResNet50 backbone models as same as the synthetic dataset experiments. Compared with bounding box precision, Inv-Mask could achieve 11.7% AP_{50} and 12.9% mAP higher than Mask R-CNN. Under the cooperation of CSB, Inv-Mask could obtains 5% AP_{50} and 17.1% mAP improvement. In addition, the SSB block could also helps Inv-Mask+CSB to improve 1.1% AP_{50} and 0.9% mAP. Compared with mask precision, Inv-Mask achieves 11.7% AP_{50} and 12.9% mAP higher than Mask R-CNN. Benefit from CSB block, Inv-Mask could obtains 17.1% AP_{50} and 12% mAP improvement. Furthermore, the proposed SSB block achieves the best mask precision at all IoU thresholds. This indicates that our methods

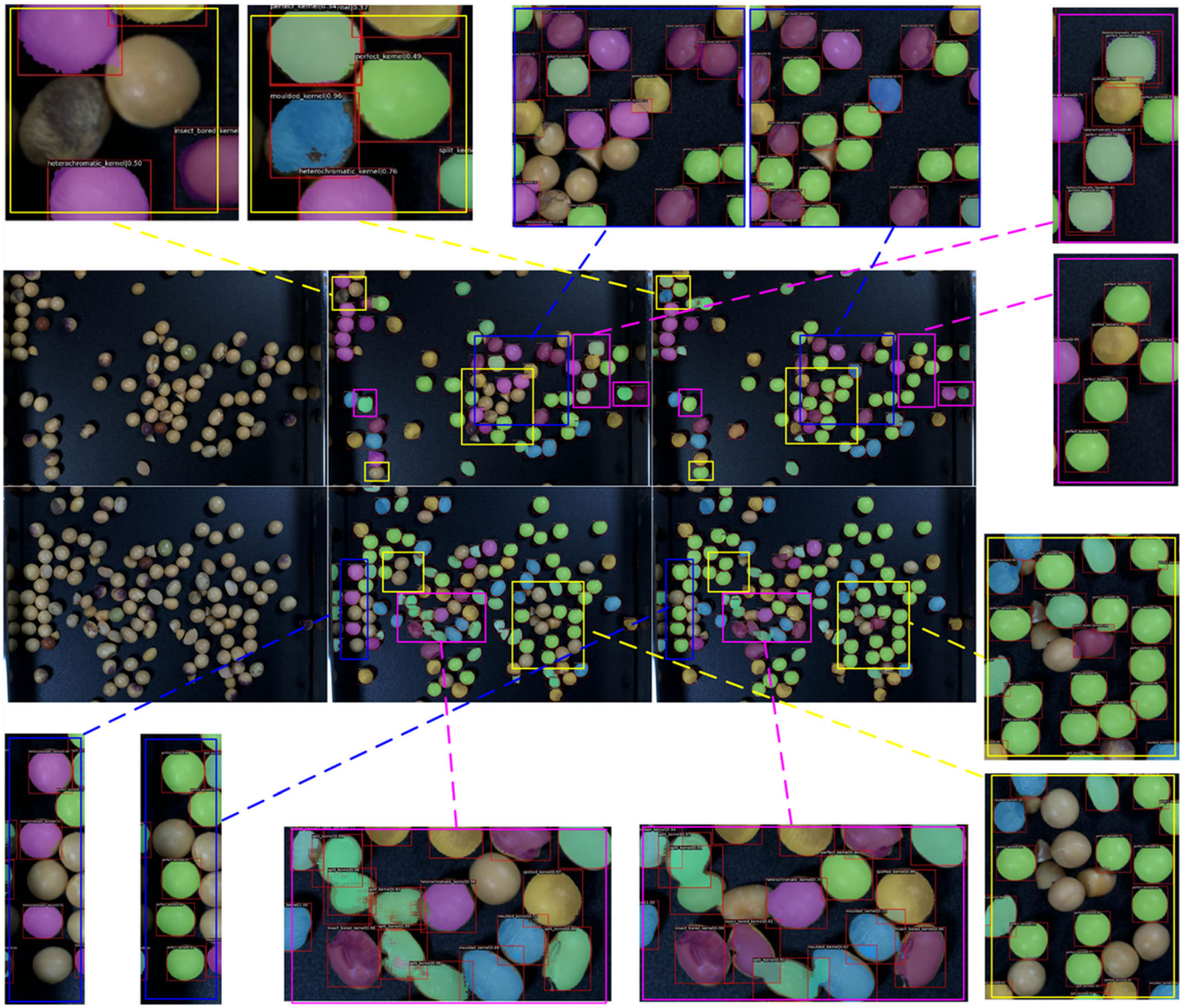

Figure 10. Real-world dataset result comparison. The left column shows the raw images, the middle column shows the results of Mask R-CNN, and the right column shows the results of our method. We enlarge some failure segmentation areas of Mask R-CNN; our model performs better in corresponding positions.

play a significant role in more precise location and segmentation.

In order to show our results more intuitively, we visualize some detection and segmentation results of real-world data in Fig. 10, with the first column showing the raw images, the second column showing the Mask R-CNN results, and the third column showing the Inv-Mask + FS-FPN results. In the enlargements of the resulting images of Mask R-CNN, the pink boxes mainly contain bounding boxes overlap occurring during detection, the blue boxes mainly include category classification errors, and the yellow boxes are cases of missed detection. In the corresponding positions, our model performs better than Mask R-CNN. Therefore, we conclude this certificates the excellent performance of our proposed model and blocks. The visualization of our experimental results is shown in Fig. 11.

To further exploit our Inv-Mask + FS-FPN, a detailed comparison of the results of each category is shown in Table 3. We notice that the broken kernels have the lowest bounding box mAP, which is caused by the small size and because they are often obscured by kernels of other categories. This also leads to poor segmentation results, and the mask mAP of 0.765 is proof of our speculation. Owing to the prominent features and large size of the moldy class, the bounding box mAP achieves 0.874. In terms of segmentation, perfect kernels have the best mask mAP, caused by the smooth surface without any damage; they can be well distinguished from other categories during segmentation.

Effectiveness of the model

We compare our methods with Mask R-CNN in Table 4. All location and segmentation results of our model outperform Mask R-CNN with lower floating-point operations (FLOPs) and parameters. Compared with Mask R-CNN, Inv-Mask reduced the FLOPs and parameters by 22.58% and 22.68% respectively; Inv-Mask + FS-FPN reduced the FLOPs and parameters by 26.51% and 29.19%

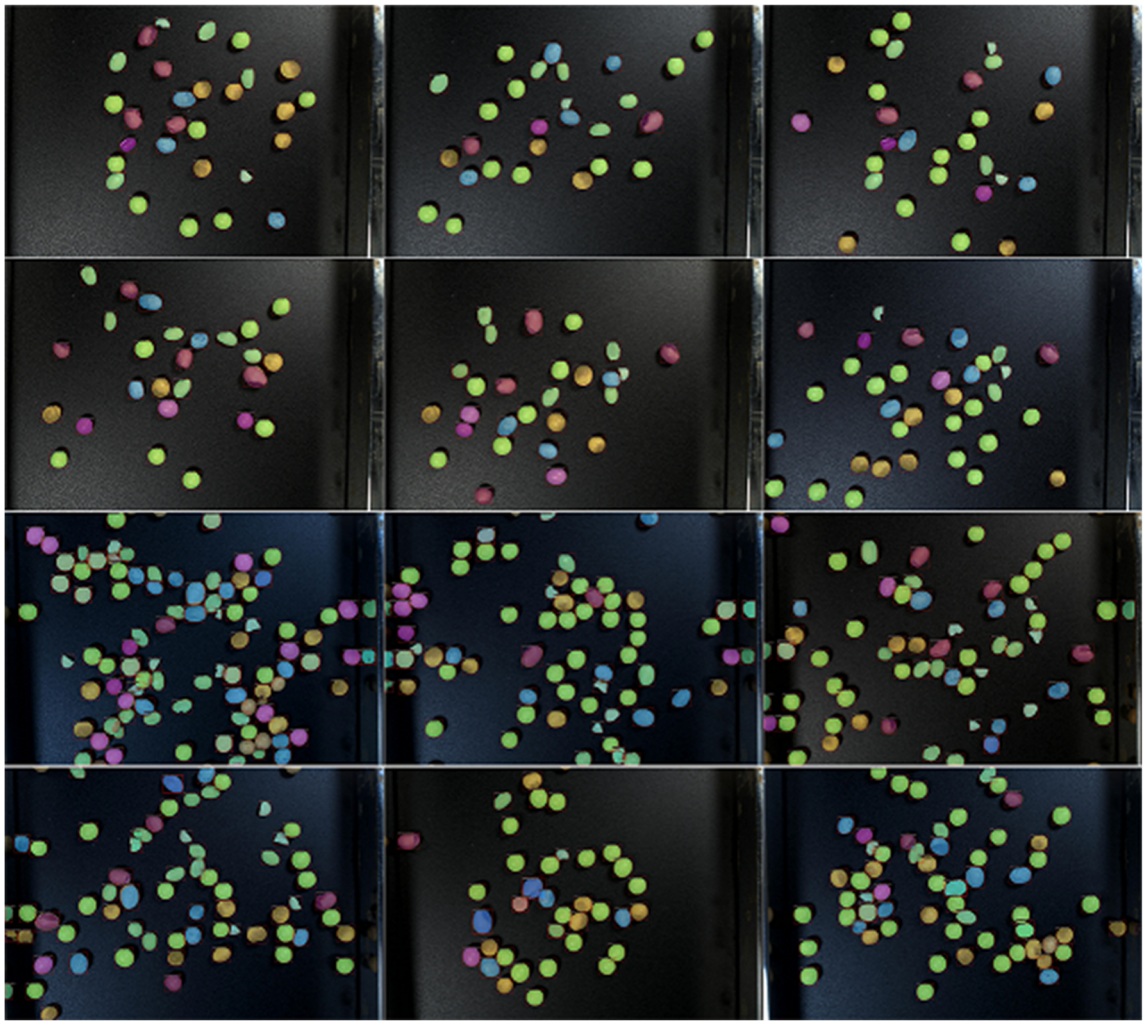

Figure 11. Visualization of the location and segmentation results using our model.

Table 3. Performance of Inv-Mask + FS-FPN on each category

Metric	Broken	Moldy	Spotted	Heterochromatic	Perfect	Insect-bored	Split
Bounding box	0.831	0.874	0.837	0.841	0.860	0.855	0.858
Mask	0.765	0.839	0.815	0.790	0.857	0.801	0.803

Table 4. Performance comparison on Mask R-CNN and our methods

Model	FLOPs ($\times 10^9$)	No. of parameters ($\times 10^6$)	AP^{bb}_{50}	mAP^{bb}	AP^{mk}_{50}	mAP^{mk}
Mask R-CNN (with FPN)	262.91	43.78	0.725	0.542	0.725	0.546
Inv-Mask (with FPN)	**203.54**	**33.85**	0.842	0.671	0.842	0.675
Inv-Mask + FS-FPN	**193.20**	**31.76**	0.903	0.851	0.903	0.810

The efficiency of our methods is greatly boosted, showing better performance than Mask R-CNN with lower cost. Bold values indicate a comparison with Mask R-CNN, highlighting the advantages of our model.
AP: average precision; bb: bounding box; FLOPs: floating-point operation\s; mAP: mean average precision; mk: mask segmentation.

respectively compared with Mask R-CNN and by 5.08% and 6.17% respectively compared with Inv-Mask. This further highlights the effectiveness of our methods, which greatly reduce the computational cost and parameter counts. The real-time performances of Mask R-CNN and our methods are demonstrated in Fig. 12. We can conclude that our methods achieve a good trade-off on detection/segmentation performance and inference time. This owes much to the characteristic of involution.

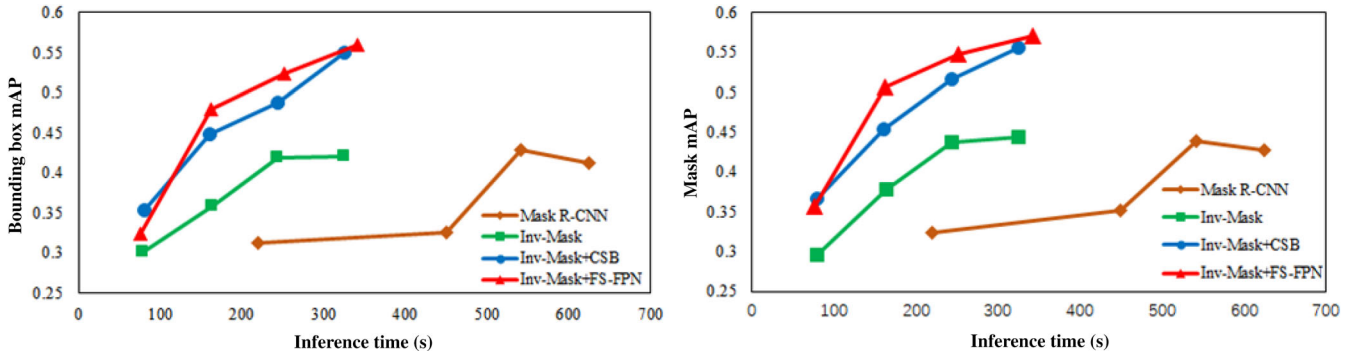

Figure 12. Inference time comparison on our methods with Mask R-CNN.

CONCLUSION

In this paper, we propose a novel self-supervision method, working to automatically generate ground-truth annotations, which can significantly reduce labor consumption. Using the self-supervised method, we prove that it can successfully prompt the neural network to analyze the real-world images of soybean seeds. Then, we present an involution-based Mask R-CNN, which is an efficient detector for locating and segmenting. Compared with the convolution-based model, the new method reverses the principles of convolution and performs better with lower computational costs. In addition, we propose an FS-FPN, which enhances the ability to extract discriminative features by involution-based FS. Experimental results demonstrate that FS-FPN could be very compatible with our Inv-Mask model. The unified network brings a significant improvement in the task of recognizing soybean kernels. In the future, we plan to further study low labor cost high-throughput seed detection methods.

ACKNOWLEDGEMENTS

This work was supported by Major Scientific and Technological Innovation Project of Shandong Province, China (2019JZZY010730).

REFERENCES

1 Kumar V, Rani A, Dixit AK, Pratap D and Bhatnagar D, A comparative assessment of total phenolic content, ferric reducing-anti-oxidative power, free radical-scavenging activity, vitamin C and isoflavones content in soybean with varying seed coat colour. *Food Res Int* **43**: 323–328 (2010). https://doi.org/10.1016/j.foodres.2009.10.019.
2 Mavridou E, Vrochidou E, Papakostas GA, Pachidis T and Kaburlasos VG, Machine vision systems in precision agriculture for crop farming. *J Imaging* **5**:89 (2019). https://doi.org/10.3390/jimaging5120089.
3 Patrício D and Rieder R, Computer vision and artificial intelligence in precision agriculture for grain crops: a systematic review. *Comput Electron Agric* **153**:69–81 (2018). https://doi.org/10.1016/j.compag.2018.08.001.
4 Lunadei L, Ruiz-Garcia L, Bodria L and Guidetti R, Image-based screening for the identification of bright greenish yellow fluorescence on pistachio nuts and cashews. *Food Bioprocess Technol* **6**:1261–1268 (2013). https://doi.org/10.1007/s11947-012-0815-8.
5 Song Y, Xie H, Ning J and Zhang Z, Grading Keemun black tea based on shape feature parameters of machine vision. *Trans Chinese Soc Agric Eng* **34**:279–286 (2018).
6 Kurtulmus F, Alibas I and Kavdir I, Classification of pepper seeds using machine vision based on neural network. *Int J Agric Biol Eng* **9**:51–62 (2016). https://doi.org/10.3965/j.ijabe.20160901.1790.
7 Olgun M, Onarcan AO, Ozkan K, Isik S, Sezer O, Ozgisi K *et al.*, Wheat grain classification by using dense SIFT features with SVM classifier. *Comput Electron Agric* **122**:185–190 (2016). https://doi.org/10.1016/j.compag.2016.01.033.
8 Momin MA, Yamamoto K, Miyamoto M, Kondo N and Grift T, Machine vision based soybean quality evaluation. *Comput Electron Agric* **140**: 452–460 (2017). https://doi.org/10.1016/j.compag.2017.06.023.
9 Tan KZ, Chai YH, Song WX and Cao XD, Identification of diseases for soybean seeds by computer vision applying BP neural network. *Int J Agric Biol Eng* **7**:43–50 (2014). https://doi.org/10.3965/j.ijabe.20140703.006.
10 Liu DJ, Ning XF, Li ZM, Yang DX, Li H and Gao LX, Discriminating and elimination of damaged soybean seeds based on image characteristics. *J Stored Prod Res* **60**:67–74 (2015). https://doi.org/10.1016/j.jspr.2014.10.001.
11 Ren SQ, He KM, Girshick R and Sun J, Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* **39**:1137–1149 (2017). https://doi.org/10.1109/tpami.2016.2577031.
12 Redmon J, Divvala S, Girshick R, Farhadi A and IEEE, You only look once: unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016).
13 Shelhamer E, Long J and Darrell T, Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell* **39**:640–651 (2017). https://doi.org/10.1109/tpami.2016.2572683.
14 Ronneberger O, Fischer P and Brox T, U-Net: convolutional networks for biomedical image segmentation. (2015).
15 Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan MX, *et al.*, Searching for MobileNetV3. Proceedings: 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019).
16 Mubin NA, Nadarajoo E, Shafri HZM and Hamedianfar A, Young and mature oil palm tree detection and counting using convolutional neural network deep learning method. *Int J Remote Sens* **40**:7500–7515 (2019). https://doi.org/10.1080/01431161.2019.1569282.
17 Wang FY, Wang RJ, Xie CJ, Yang P and Liu L, Fusing multi-scale context-aware information representation for automatic in-field pest detection and recognition. *Comput Electron Agric* **169**:105222 (2020). https://doi.org/10.1016/j.compag.2020.105222.
18 Li R, Wang RJ, Xie CJ, Liu L, Zhang J, Wang FY *et al.*, A coarse-to-fine network for aphid recognition and detection in the field. *Biosyst Eng* **187**: 39–52 (2019). https://doi.org/10.1016/j.biosystemseng.2019.08.013.
19 Veeramani B, Raymond JW and Chanda P, DeepSort: deep convolutional networks for sorting haploid maize seeds. *BMC Bioinformatics* **19**:289 (2018). https://doi.org/10.1186/s12859-018-2267-2.
20 Steinbrener J, Posch K and Leitner R, Hyperspectral fruit and vegetable classification using convolutional neural networks. *Comput Electron Agric* **162**:364–372 (2019). https://doi.org/10.1016/j.compag.2019.04.019.
21 Bosilj P, Aptoula E, Duckett T and Cielniak G, Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *J Field Rob* **37**:7–19 (2020). https://doi.org/10.1002/rob.21869.
22 Misra T, Arora A, Marwaha S, Chinnusamy V, Rao AR, Jain R *et al.*, SpikeSegNet – a deep learning approach utilizing encoder–decoder network with hourglass for spike segmentation and counting in wheat plant from visual imaging. *Plant Methods* **16**:40 (2020). https://doi.org/10.1186/s13007-020-00582-9.
23 Ardizzone, L., J. Kruse, S.J. Wirkert, D. Rahner, E. Pellegrini, R.S. Klessen, et al., Analyzing inverse problems with invertible neural networks. arXiv:1808.04730 (2019).

24 Gomez, A.N., M. Ren, R. Urtasun and R.B. Grosse, The reversible residual network: backpropagation without storing activations. arXiv: 1707.04585 (2017).
25 Varol G, Romero J, Martin X, Mahmood N, Black MJ, Laptev I, *et al*., Learning from synthetic humans. Proceedings: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017).
26 Ward, D., P. Moghadam and N. Hudson. Deep leaf segmentation using synthetic data. arXiv:1807.10931 (2018).
27 He KM, Gkioxari G, Dollar P, Girshick R and Mask R-CNN, Proceedings: 2017 International Conference on Computer Vision. (2017).
28 Russell BC, Torralba A, Murphy KP and Freeman WT, LabelMe: a database and web-based tool for image annotation. *Int J Comput Vis* **77**:157–173 (2008). https://doi.org/10.1007/s11263-007-0090-8.
29 Maclaurin D, Duvenaud D and Adams RP. Gradient-based hyperparameter optimization through reversible learning. ICML'15: Proceedings of the 32nd International Conference on Machine Learning (2015).
30 Dinh, L., D. Krueger and Y. Bengio, NICE: non-linear independent components estimation. arXiv:1410.8516 (2014).
31 Dinh, L., J. Sohl-Dickstein and S. Bengio, Density estimation using real NVP. arXiv:1605.08803 abs/1605.08803 (2017).
32 He KM, Zhang XY, Ren SQ, Sun J and IEEE, Deep residual learning for image recognition, in *Proceedings: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, pp. 770–778 (2016).
33 Jie H, Li S, Gang S and Albanie S, Squeeze-and-excitation networks. *IEEE Trans Pattern Anal Mach Intell* **42**:2011–2023 (2020). https://doi.org/10.1109/tpami.2019.2913372.

