SPCN: An Innovative Soybean Pod Counting Network Based on HDC Strategy and Attention Mechanism

Ximing Li¹,², Yitao Zhuang¹, Jingye Li¹, Yue Zhang¹, Zhe Wang¹, Jiangsan Zhao³, Dazhi Li¹ and Yuefang Gao¹,*

¹ College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China; liximing@scau.edu.cn (X.L.); 20222157014@stu.scau.edu.cn (Y.Z.); wangzhe@stu.scau.edu.cn (Z.W.)
² Key Laboratory of Smart Agricultural Technology in Tropical South China, Ministry of Agriculture and Rural Affairs, Guangzhou 510642, China
³ Department of Agricultural Technology, Norwegian Institute of Bioeconomy Research (NIBIO), P.O. Box 115, NO-1431 Ås, Norway; jiangsan.zhao@nibio.no
* Correspondence: gaoyuefang@scau.edu.cn

Abstract: Soybean pod count is a crucial aspect of soybean plant phenotyping, offering valuable reference information for breeding and planting management. Traditional manual counting methods are not only costly but also prone to errors. Existing detection-based soybean pod counting methods face challenges due to the crowded and uneven distribution of soybean pods on the plants. To tackle this issue, we propose a Soybean Pod Counting Network (SPCN) for accurate soybean pod counting. SPCN is a density map-based architecture based on Hybrid Dilated Convolution (HDC) strategy and attention mechanism for feature extraction, using the Unbalanced Optimal Transport (UOT) loss function for supervising density map generation. Additionally, we introduce a new diverse dataset, BeanCount-1500, comprising of 24,684 images of 316 soybean varieties with various backgrounds and lighting conditions. Extensive experiments on BeanCount-1500 demonstrate the advantages of SPCN in soybean pod counting with an Mean Absolute Error (MAE) and an Mean Squared Error (MSE) of 4.37 and 6.45, respectively, significantly outperforming the current competing method by a substantial margin. Its excellent performance on the Renshou2021 dataset further confirms its outstanding generalization potential. Overall, the proposed method can provide technical support for intelligent breeding and planting management of soybean, promoting the digital and precise management of agriculture in general.

Keywords: computer vision; density map-based counting; soybean pod; hybrid dilated convolution; Convolutional Block Attention Module

1. Introduction

Soybeans are one of the most important sources of protein in the world [1]. Consequently, accelerating the breeding and cultivation of new high-yield, high-quality, and disease-resistant varieties is essential. Previous studies have indicated that the number of pods is a crucial factor in soybean yield estimation, providing vital information for plant breeding [2–4]. However, traditional manual counting methods are time-consuming and labor-intensive, particularly in large-scale cultivation scenarios where the workload becomes immense and unmanageable. Moreover, manual counting is susceptible to subjective errors, undermining the accuracies of decisions and assessments. These challenges not only diminish the management efficiency of agricultural practices but also ruin the scientific efforts in precision breeding [5–7].

Computer vision, as a non-invasive, rapid, and objective technique, holds significant promise across various domains, including in soybean pod counting. Through automated image analysis, computer vision technology can effectively lower labor costs and enhance the speed and accuracy of the counting procedure. In recent years, the emergence and
advancement of deep learning techniques has notably propelled their applications in vision-based crop counting methods [8–12]. Object detection-based counting methods identify objects in a digital image or video first and then count the detected objects, which are usually in the form of bounding boxes. It has been predominantly employed in recent studies for different counting tasks. Riera et al. [13] introduce a deep learning-based multi-view image fusion framework tailored for processing and analyzing multi-angle videos obtained by ground robots in fields to accurately count soybean pods for yield estimation. Xu et al. [14] propose a VFNet detector-based Deformable Attention Recursive Feature Pyramid Network for Soybean Pod Counting (DARFP-SD). Mathew et al. [15] investigate the application of object detection models, both YOLOV7 and YOLOv7-E6E, for soybean pod counting. Xiang [16] proposed YOLO POD, a method based on the YOLO X framework, to achieve accurate identification and counting of soybean pods without compromising inference speed. He et al. [17] propose a YOLOV5-based soybean pods recognition and weight estimation model to achieve accurate estimation pod weight of a single plant. Yu et al. [3] introduce PodNet for accurate counting and localizing soybean pods with a minimized number of parameters. However, object detection-based counting methods struggle in scenarios characterized by high density and substantial occlusion. On the contrary, counting objects based on a density map can circumvent the necessity of pinpointing objects’ exact locations, and thereby achieving higher prediction accuracies [18–20].

The density map-based counting method entails estimating the number of objects by analyzing the density distribution within an image. The density map method estimates the precise count of objects through mapping the extracted features to the density map. This approach has been widely applied in various scenarios of densely located objects, such as crowd counting [18–21] and Cell counting [22–24], etc. In the context of soybean pod counting, the research scenario is plagued by challenges like mutual occlusion and uneven density distribution of soybean pods, akin to the complexities encountered in crowd counting scenarios. This suggests that the density map-based counting method is inherently well-suited for soybean pod counting. However, the potential of density map-based soybean pod counting methods have been relatively less well studied.

Therefore, this paper introduces a novel density map-based pod counting framework, named Soybean Pod Counting Network (SPCN). SPCN integrates both the attention mechanism and the hybrid dilated convolution (HDC) technique to enhance counting accuracy, effectively addressing pod occlusion by using spatial contextual information in crowded regions. Additionally, the study introduces an Unbalanced Optimal Transport (UOT) based loss function for direct density map generation to further enhance counting accuracy and system robustness. To demonstrate the effectiveness of the proposed method, it is tested not only on our large-scale soybean pod counting dataset, BeanCount-1500, but also on two other publicly available counting datasets.

The contributions of this paper are summarized as follows:

Proposed SPCN: This paper pioneers the density map-based SPCN architecture, which notably enhances the accuracy and efficiency of soybean pod counting by integrating the attention mechanism with the HDC strategy and supervising density map generation using the UOT loss function. The source code is available at https://github.com/johnhamtom/soybean_counting_SPCN (accessed on 10 August 2024).

Construction of BeanCount-1500 dataset: We constructed the large-scale BeanCount-1500 soybean pod counting dataset, encompassing diverse lighting conditions, backgrounds, clarity levels, occupation level between plant and pod, as well as instances of pod shatter, providing a valuable resource for soybean pod counting research.

2. Materials and Methods
2.1. Data Acquisition and Preprocessing

To meet the breeding needs of the soybean breeding team at the College of Agriculture, South China Agricultural University, it is necessary to accurately count the number of soybean pods of different soybean varieties. In collaboration with the College of Agricul-
ture, we collected a large number of images of genetically sequenced soybean plants and developed a deep network model to achieve robust automatic soybean pod counting.

Specifically, the collected data covers 316 soybean varieties from different countries of origins. They were all planted at the planting base of South China Agricultural University in June 2019 and harvested in October the same year for image collection. We used various mobile devices, including the HUAWEI Mate 20 (Shenzhen, Guangdong, China) and iPhone 13 (Cupertino, CA, USA), to capture images of these soybean plants. The images were taken under natural light conditions, with backgrounds set in both a clean black curtain environment and an open, noisy environment. For each variety, we randomly selected 4–6 samples and took 10–15 images of each sample from different angles, then manually counted the number of pods on each soybean plant. After thoroughly cleaning and annotating the collected data, we created a high-quality dataset named Bean-Count-1500. Figure 1 shows some example images from Bean-Count-1500.

Figure 1. Sample images of BeanCount-1500.

As illustrated in Figure 2a, during image acquisition, two categories were defined: one category features simple scenes, where soybean plants were photographed against a simple background such as a black backdrop or a clear wall; the other category is characterized by complex real scenes involving authentic backgrounds like fields or corridors. In the situation of complex real scenes, images also vary in levels of clarity, captured by jittering or defocusing the camera, as exemplified in Figure 2b. Additionally, soybean images are categorized into single-plant and multi-plant images, as shown in Figure 2c. Mutual shading caused by dense pod clusters (shattering (Figure 2d), together with these image diversities mentioned above, create extremely challenging situations for accurate pod counting.
To ensure the accuracy of the data, we invited experienced technicians to annotate pods with points inside the images using the labelme tool [25] and to register the number of pods per soybean plant at the same time. For point annotation, a point is put at the center of the visible part of each pod to make sure representative features can be extracted from the pod. Images of soybean plants with no clear pod outlines are deleted during annotation. In addition, cases of pod shatter and partial pod occlusion are also included to cover the diversity of real-world scenarios. As shown in Figure 3, pods whose shape can be shown completely in the picture are labelled with the point right at the center, while pods that are slightly occluded but still show approximate outlines are still labelled with the point close to the center. The complete annotated dataset, BeanCount-1500, contains a total of 24,684 accurately labelled images of 1500 soybean plants with a total of 987,391 pods across 316 varieties. These images of soybean pods are taken under varying conditions, such as different light, background, clarity, number of plants, and pod shatter, to facilitate building a more robust soybean pod counting model.

We randomly sampled 16,456 soybean pod images from the BeanCount-1500 dataset for training. A validation set comprising of 4113 images was used for model selection. Another set of 4115 soybean pod images was used to evaluate the model’s final performance (Table 1).
Table 1. The details of BeanCount-1500.

<table>
<thead>
<tr>
<th></th>
<th>Number of Pictures</th>
<th>Total Number of Labels</th>
<th>Median Number of Labels</th>
<th>Average Number of Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set</td>
<td>16,456</td>
<td>658,033</td>
<td>36</td>
<td>39,987</td>
</tr>
<tr>
<td>validation set</td>
<td>4113</td>
<td>164,341</td>
<td>36</td>
<td>40,121</td>
</tr>
<tr>
<td>test set</td>
<td>4115</td>
<td>165,017</td>
<td>36</td>
<td>39,937</td>
</tr>
</tbody>
</table>

To elucidate the characteristics of the BeanCount-1500 dataset more comprehensively, we employ three visualization techniques: three-dimensional scatterplots, box plots, and two-dimensional scatterplots. First, Figure 4a depicts a three-dimensional scatter plot of BeanCount-1500, with the x-axis, y-axis, and z-axis representing image width (w), height (h), and the number of pod labels, respectively. The lighter the color, the larger the number of pods per image. This plot illustrates the relationship between pod label count and image resolution, revealing discernible trends and distribution patterns. Figure 4b presents a box-and-line plot demonstrating the distribution of pod annotations across the dataset. The plot highlights that most images exhibit approximately 50 pod labels, alongside a notable of images either with no pod or densely populated images exceeding 150 pods per image. Additionally, Figure 4c showcases a two-dimensional scatter plot of image resolutions categorized into three groups using K-means clustering, represented by purple, yellow, and cyan dots. The analysis indicates that 11,844 images are in resolutions ranging from $512 \times 512$ to $1728 \times 1376$ (yellow dots), 9708 images in resolutions spanning $512 \times 1129$ to $1024 \times 4348$ (purple dots), and 3132 images in resolutions ranging from $1690 \times 512$ to $3795 \times 1024$ (cyan dots).
This paper introduces a novel model architecture named SPCN, which is a dilated convolution network incorporating an attention module. The architecture, depicted in Figure 5, can be divided into three main modules: the front-end module, the attention module, and the back-end module.

The front-end module primarily involves a series of stacked convolutional and pooling layers serving as 2D feature extraction. The CBAM (Convolutional Block Attention Module) module enhances the extracted features by introducing both channel attention and spatial attention mechanisms, effectively weighting and focusing on more important features before forwarding them to the back-end network. The back-end module employs a set of extended dilated convolutions to expand the receptive field. Finally, the output is supervised using the UOT loss function to guide density map generation. This comprehensive
architecture facilitates accurate soybean pod counting by using advanced techniques in feature extraction, attention mechanisms, and loss function supervision.

Figure 5. Structure of the SPCN.

2.2.1. Front-End Module

As depicted in Figure 6, using the exceptional feature extraction capabilities of the VGG19 network [26], we discard the last layers of VGG19, including a max pooling, an adaptive average pooling, three Fully Connected Layer, and one softmax layer, and treat the remaining layers as the core of the front-end module. Specifically, the module employs an up-sampling operation to increase the resolution of the feature map before being passed to the attention module. In this process, the convolutional layers are responsible for mining the deep features of the input image, and the up-sampling operation enlarges the size of the feature map, which eventually captures rich information more effectively.

Figure 6. Illustration of the Front-End module.
In conclusion, the design of this front-end network structure demonstrates a high degree of flexibility and extensibility, making it easy to adjust or replace according to specific needs in different application scenarios.

2.2.2. CBAM Module

We used the CBAM Attention Module [27] to increase the model performance. CBAM has been widely used in deep learning to improve feature representation power. It is used to focus both the channel and spatial information of the input feature map. The channel attention module improves the attention to important features by adaptively weighting each channel. The spatial attention module, on the other hand, weights features at different locations differently to improve the model’s attention to important regions. The combination of these two sub-modules enables the model to better focus on more meaningful information from the input features to improve both the accuracy and generalization potential of the derived model.

For a sheet of input features, channel attention and spatial attention can compute complementary attentions focusing on ‘what’ and ‘where’, respectively. Furthermore, it has been shown that sequential arrangements can give better results than parallel arrangements, and that channel attention is slightly better prioritized than spatial attention [27]. As shown in Figure 7, the input feature map F is first channel-weighted by channel attention and then spatially weighted by spatial attention.

![Figure 7. The structure of CBAM.](image)

2.2.3. Back-End Module

Dilated convolution, a special convolution operation, introduces additional spacing within a conventional convolution kernel, thereby expanding the effective receptive field without inflating parameters or computational load. This technique enables the model to capture a broader range of contextual information without sacrificing resolution. However, dilated convolution’s spaced sampling of the feature map can lead to a grid effect, causing loss of feature information. Drawing inspiration from Wang et al. [28], this paper employs the HDC strategy in the back-end module, which involves a strategic combination of dilated convolutions with varying dilation rates to mitigate the grid effect and expand the receptive field.

As illustrated in Figure 8a, a 3 × 3 convolution kernel with a dilation rate of 1 is depicted in the upper left corner, while a kernel with a dilation rate of 2 is shown in the upper right corner. When processed by three convolutional kernels with a size of 3 × 3 and a dilation rate of [2,2,2], the receptive field surrounding the red pixel point in the center appears grid-like. However, with a dilation rate of [1–3], the receptive field is more evenly distributed.
In our work, we employ a Generalized Loss function for supervised density map generation based on UOT, as proposed by Wan et al. [29]. This loss function enables the trained network model to generate high-quality density maps by minimizing the direct transport cost between the predicted density maps and the ground truth (GT) point annotations. Additionally, L1-loss and L2-loss are also used to measure the difference between predicted density map and the GT point annotations, as illustrated in Figure 9. This approach ensures that the trained network model produces high-quality density maps that exhibit desirable characteristics, such as more ‘sparse’ and more precise localization.

In this paper, \( a = [a_i] \) and \( b = [b_j] \) are defined as the predicted density map and GT point map, respectively. The General loss formula is shown in (1).

\[
L^T_C(A, B) = \min_{P \in R_{s+m}^{n \times m}} < C, P > - \epsilon H(P) + \tau D_1(P1_m | a) + \tau D_2(P^T1_n | b) \tag{1}
\]

The loss function can be divided into four parts, the first part being the \( \min_{P \in R_{s+m}^{n \times m}} < C, P > \). \( C \in R_{s+m}^{n \times m} \) It is the transmission cost matrix, where \( C_{ij} = C(x_i, y_j) = \exp \left( \frac{1}{\eta} \frac{1}{\eta(x_i, y_j)} \right) \), which refers to the transport cost to move density map of \( x_i \) to the GT point coordinates \( y_j \). \( \eta(x_i, y_j) \) is an adaptive perspective factor that is mapped between \( (1/2) \ast (h_{x_i} + h_y \}, \) this can make the density generation of distant pods in Bean-Count1500 sparser for some images with perspective effects. \( P \) is the transport matrix and has a unique solution and a unique form [30]. It determines the optimal transfer scheme between density map pixels and GT points to minimize the transfer cost. By minimizing the transmission cost, it is possible to encourage the prediction of density values in the
vicinity of the annotations, pushing the predicted density towards the GT point annotations during training.

![Diagram](image)

**Figure 9.** Schematic diagram of the loss function.

The second part $\varepsilon H(P) = -\varepsilon \sum_{ij} P_{ij} \log P_{ij}$ is the entropy regularisation term, which calculates the degree of overlap of the transmission matrices in order to reduce the ineffective transmission of the transmission matrices and ultimately bring the predicted density values closer to the GT points. $\varepsilon$: the larger this, the less compact the density map predicted by the network model.

The third part $D_1(P_{1m} | a)$ is the pixel-by-pixel loss between the predicted density map $a$ and the intermediate density map $\tilde{a} = P_{1m}$ constructed based on the transfer matrix $P$ and GT point coordinates. In this paper, we use the L2 loss to measure the difference between them (see Equation (2)). This term allows any predicted density that is not associated with an annotation to be penalized.

$$D_1(P_{1m} | a) = \| P_{1m} - a \|^2_2$$ (2)

The fourth part $D_2(P^T_{1n} | b)$ is the point-by-point loss between the annotations constructed $\tilde{b} = P^T_{1n}$ using the transmission matrix $P$ and the predicted density maps and the GT point annotations $b$. In this paper, the L1 loss is used to measure the gap between them (see Equation (3)). This item ensures that all GT point annotations are taken into account, i.e., used in the transmission scheme.

$$D_2(P^T_{1n} | b) = \| P^T_{1n} - b \|_1$$ (3)

The final General loss supervised trained network model outputs a predicted density map that is summed over the pixel values (see Equation (4), where $D^{est}(x)$ is the predicted density map and $W$ and $L$ are the width and height of the predicted density map, respectively, used to calculate the number of predicted pods.

$$\text{Count} = \sum_{w=1}^{W} \sum_{l=1}^{L} D^{est}(x)$$ (4)

### 2.3. Algorithm Implementation, Parameter Setting, and Evaluation Metrics

In summary, the feature extraction module of the SPCN can be outlined in three key steps: first, the input image undergoes feature extraction and spatial up-sampling through the front-end network structure. Next, the representation power of extracted features are improved by the CBAM module. Finally, the receptive field is expanded through dilated
convolution for more contextual information extraction in the back-end network. The resulting output density map is then utilized for counting.

In the model, the SPCN architecture is initially configured and trained using a specified number of epochs, datasets, and batch sizes within the training module. Subsequently, once the target image completes the feature extraction process, the resulting output density map is passed to the generalized loss function for density estimation and counting. The pseudo-code outlining this process is presented in Algorithm 1.

**Algorithm 1: Training Module**

```
Input: w×h×3 Image of bean dataset
Output: Output feature density map of 1 channel
Model = (SPCN)// Initialize the model as SPCN
Train (Epochs):
   Model.train(dataset, batch size)
   //Feature extraction part
   For x in dataset
      Input == x
      x = Front−end(x): feat = [64×2,M,128×3,M,256×4,512×4](M == Maxpooling)
      x = CBAM(x): feat = [512]
      x=Back−end(x): feat = [256,128,64]
      x = output_layer(x): channel = 1
      Return x
   // Loss function computation
   Loss = Generalized Loss (Model.train(output))
   Expected calculation, print MAE, MSE
```

The code builds and experiments were performed in the runtime environment shown in the Table 2.

**Table 2. Operating environment table.**

<table>
<thead>
<tr>
<th>Software and Drivers</th>
<th>System: Windows 10</th>
<th>CPU: Intel(R)12GenCore i7-12700KF 3.6 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pytorch: 1.10.1</td>
<td>GPU: GeForce RTX 3060Ti</td>
</tr>
<tr>
<td></td>
<td>CUDA:11.3+CUDNN:8.2</td>
<td>Memory: 16 GB</td>
</tr>
<tr>
<td></td>
<td>Python: 3.8.15</td>
<td>Disk: 3 TB</td>
</tr>
</tbody>
</table>

To assess the model’s performance, we employed two evaluation metrics, Mean Absolute Error (MAE) and Mean Squared Error (MSE), on the test dataset. These metrics effectively gauge the model’s accuracy and robustness, respectively, offering valuable insights for further refinement. The formulas for MAE and MSE are as follows, where \( y_i \) represents the true value of pods in the ith picture and \( \hat{y}_i \) represents the predicted value of pods in the ith picture.

\[
    MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{5}
\]

\[
    MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{6}
\]

**3. Results**

**3.1. Training Details**

The model implemented in this paper is built upon the PyTorch framework. To enhance training efficacy, we employed the Adam optimizer with an initial learning rate set to \( 1 \times 10^{-5} \), a weight decay of 0.0001, and a momentum parameter of 0.9.
During the training process, we utilized a batch size of 1 and conducted 100 training epochs for soybean pod counting training. We set validation to start at epoch 10 of training and to occur every 2 epochs thereafter. Given the extensive size of the BeanCount-1500 dataset, the SPCN sufficiently converged after 100 training iterations, as depicted in Figure 10. Additionally, the MAE and MSE on the training and validation sets gradually converge during the training process, as shown in Figure 11.

![Figure 10. Decreasing training loss graph.](image)

![Figure 11. Decreasing training and validation MAE and MSE graph.](image)

3.2. Comparative Experimental Results

Table 3 shows the counting performances of three models, original GL [29], GL with feature extraction module from CSRNet [19], and SPCN, on our BeanCount-1500 dataset. VGG19 is used in GL as the feature extraction network while the feature extraction module in CSRNet is composed of multiple stacked dilated convolutions. In addition, GL also uses Generalized Loss function as the loss function.

![Figure 10. Decreasing training loss graph.](image)

![Figure 11. Decreasing training and validation MAE and MSE graph.](image)
Table 3. Comparison of soybean pod counting experiments. Bold numbers indicate the best-performing metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL</td>
<td>4.89</td>
<td>7.23</td>
</tr>
<tr>
<td>GL (CSRNet)</td>
<td>5.06</td>
<td>7.45</td>
</tr>
<tr>
<td>Ours (SPCN)</td>
<td>4.37</td>
<td>6.45</td>
</tr>
</tbody>
</table>

As can be seen from Table 3, SPCN achieves the highest performance with the lowest values of 4.37 in MAE and 6.45 in MSE. GL (CSRNet) has the worst performance, which is due to the gridding issue caused by a large number of dilated convolutions stacked with the fixed dilation rate, resulting in a huge feature information loss.

Three randomly selected images of soybean pods with varying degrees of densities are presented in Figure 12. In Figure 12, “Origin” denotes the input image, while “GL”, “GL (CSRNet)”, and “SPCN” represent different models. Upon visual inspection, it is evident that the predicted and actual values of the density map generated by the proposed SPCN aligns better with the point label compared to the other two models. One of the notable strengths of the SPCN model is demonstrated by predictions inside the yellow circle. For instance, in IMG_2262, the SPCN model successfully detects a pod that closely resembles a branch in the yellow circle region, which is unreachable by either GL or GL (CSRNet). Similarly, in IMG_10481, where numerous pods overlap with leaves in the yellow circle part, the SPCN model exhibits superior detection capabilities compared to its counterparts. In IMG_10726, the region highlighted by the yellow circle with overlapped pods and yellow leaves poses a considerable challenge to most of counting models. GL experiences a high misdetection rate and GL (CSRNet) generates a blurred density map, whereas our proposed SPCN model successfully generates a high-quality density map.

Figure 12. Visualization of model predictions. The first column shows the input image with the actual number of pods in the upper right corner. The second to fourth columns display the density maps predicted by different models with the predicted number of pods in the upper right corner. The fifth column shows the distribution of annotation points. Yellow circles highlight areas where the SPCN model performs better.

Another set of experiments compare SPCN with different crowd counting models, including the classical model CSRNet [19] and HRNet [31] and models with different loss
functions; the BL [20] and DM-Count [32], CCTrans [33], and BCCMA [21] are shown in Table 4. We used the default optimal parameters of these open-source models for comparisons in our experiments. As can be seen in Table 4, all models compared in the study have worse performances than our proposed SPCN in soybean pod counting, highlighting the superiority of SPCN.

### Table 4. Experimental comparison with mainstream counting models. Bold numbers indicate the best-performing metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSRNet</td>
<td>16.29</td>
<td>21.89</td>
</tr>
<tr>
<td>BL</td>
<td>6.11</td>
<td>8.94</td>
</tr>
<tr>
<td>HRNet</td>
<td>32.90</td>
<td>41.41</td>
</tr>
<tr>
<td>BCCMA</td>
<td>39.94</td>
<td>47.98</td>
</tr>
<tr>
<td>DM-Count</td>
<td>7.26</td>
<td>10.76</td>
</tr>
<tr>
<td>GL</td>
<td>4.89</td>
<td>7.23</td>
</tr>
<tr>
<td>CCTrans</td>
<td>4.91</td>
<td>7.28</td>
</tr>
<tr>
<td>Ours (SPCN)</td>
<td><strong>4.37</strong></td>
<td><strong>6.45</strong></td>
</tr>
</tbody>
</table>

### 3.3. Analysis of Ablation Experiments

To further analyze the contribution of different modules to the model’s better performance, we conducted ablation studies on different modules of SPCN.

In comparison to models utilizing the same Generalized Loss (GL) loss function, SPCN outperforms GL(VGG19), GL(HRNet), and GL(CSRNet) models with various feature extraction modules.

In addition, the CSRNet version D utilized in this study is characterized by a backend comprising six convolutional layers with a kernel size of $3 \times 3$ and a dilation rate of 4. To validate the effectiveness of the HDC strategy, the dilation rates of the six $3 \times 3$ convolutional kernels in the backend are adjusted from $[4,4,4,4,4,4]$ to $[1–3,5,7,7]$. This adjustment mitigates grid effects and still preserves the same receptive field size, denoted as CSRNetHDC. The lower MAE and MSE results stemmed from the adjustment over the HDC strategy, demonstrating the effectiveness of the HDC strategy. The models employed and the corresponding experimental results are presented in Table 5.

### Table 5. Comparison of ablation experiment 1. Bold numbers indicate the best-performing metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL</td>
<td>4.89</td>
<td>7.23</td>
</tr>
<tr>
<td>GL (HRNet)</td>
<td>21.41</td>
<td>28.60</td>
</tr>
<tr>
<td>GL (CSRNet)</td>
<td>5.06</td>
<td>7.45</td>
</tr>
<tr>
<td>GL (CSRNetHDC)</td>
<td>4.91</td>
<td>7.24</td>
</tr>
<tr>
<td>Ours (SPCN)</td>
<td><strong>4.37</strong></td>
<td><strong>6.45</strong></td>
</tr>
</tbody>
</table>

As illustrated in Table 6, the MAE and MSE of the GL model in soybean pod counting are 4.89 and 7.23, respectively. The integration of CABM reduces MAE and MSE by 0.38 and 0.50, respectively. The incorporation of both CBAM and HDC design reaches the most optimal performance with MAE and MSE of 4.32 and 6.33, respectively, which substantiates the efficacy of the SPCN design.

### Table 6. Comparison of ablation experiment 2. Bold numbers indicate the best-performing metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL</td>
<td>4.89</td>
<td>7.23</td>
</tr>
<tr>
<td>With CBAM</td>
<td>4.51</td>
<td>6.73</td>
</tr>
<tr>
<td>With HDC</td>
<td>4.45</td>
<td>6.57</td>
</tr>
<tr>
<td>With CBAM and HDC</td>
<td><strong>4.37</strong></td>
<td><strong>6.45</strong></td>
</tr>
</tbody>
</table>
Finally, we conducted ablation studies on the HDC strategy by comparing three arrangements for the back-end module. Arrangement 1 entails the change of a dilation rate to 2, resulting in a similar receptive field as the original Back-End module, but suffering from the extra grid effect. Arrangement 2 involves convolutional layers with a fixed dilation rate of 3, which leads to a larger receptive field than that of strategy 3 but suffers from the grid effect when compared to the Back-End module in SPCN. Table 7 demonstrates the superior effectiveness of the arrangement of dilated convolution layers within the Back-End module of SPCN, circumventing the grid effect and reaching higher counting accuracy.

Table 7. Comparison of ablation experiment 3. Bold numbers indicate the best-performing metrics.

<table>
<thead>
<tr>
<th>Methodologies</th>
<th>MAE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy 1: [2,2,2]</td>
<td>4.54</td>
<td>6.70</td>
</tr>
<tr>
<td>Strategy 2: [3,3,3]</td>
<td>4.45</td>
<td>6.53</td>
</tr>
<tr>
<td>Strategy 3: [1,2,3]</td>
<td>4.37</td>
<td>6.45</td>
</tr>
</tbody>
</table>

In order to verify the generalization ability of the SPCN model in different soybean pod counting scenarios, we trained the SPCN model on our proposed BeanCount-1500 dataset and tested the open source soybean pod counting datasets [16]. The first dataset is the Chongzhou dataset, composed of 570 images captured by a Canon 700D with resolution of $4752 \times 3168$ pixels at the Chongzhou Experimental Base in Sichuan Agricultural University in 2021. The second is the Renshou2021 dataset, comprising 878 images taken by a Canon 750D with a resolution of $5184 \times 2196$ pixels at Sichuan Agricultural University’s Renshou Farm in 2021.

In this paper, we used the training set of BeanCount-1500 to train the SPCN model; 878 images from the Renshou2021 dataset are used as the test set, maintaining the same test benchmarks as the YOLO POD model [16] and PodNet model [3]. Except for SPCN, all models are trained on the Chongzhou dataset. As can be seen from Table 8, the SPCN model shows strong generalization capability, with an MAE of 4.43, which is only 0.25 away from the optimal MAE, and an MSE of 6.18, which is the current optimal level. Figure 13 shows four display plots of the test results; it is clear that the SPCN model is able to accurately capture pod features and provide excellent counting accuracy. Although the test results on the Renshou2021 dataset are slightly inferior to those of YOLO POD and PodNet, SPCN performs better in terms of MSE.

Table 8. Performance of the proposed method on the Renshou2021 dataset. Bold numbers indicate the best-performing metrics.

<table>
<thead>
<tr>
<th>Methodologies</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv8 [3,34]</td>
<td>6.16</td>
<td>8.89</td>
</tr>
<tr>
<td>CenterNet [3,35]</td>
<td>12.67</td>
<td>17.77</td>
</tr>
<tr>
<td>Faster R-CNN [3,36]</td>
<td>16.56</td>
<td>19.22</td>
</tr>
<tr>
<td>TasselLFANet [3,37]</td>
<td>5.92</td>
<td>8.83</td>
</tr>
<tr>
<td>EfficientDet [3,38]</td>
<td>22.74</td>
<td>38.41</td>
</tr>
<tr>
<td>FCOS [3,39]</td>
<td>11.88</td>
<td>22.00</td>
</tr>
<tr>
<td>YOLO POD [16]</td>
<td>4.18</td>
<td>10.04</td>
</tr>
<tr>
<td>PodNet [3]</td>
<td>4.52</td>
<td>7.65</td>
</tr>
<tr>
<td>SPCN (train on BeanCount-1500)</td>
<td>4.43</td>
<td>6.18</td>
</tr>
</tbody>
</table>
4. Discussion

Compared with the mainstream method [19–21,31–33], SPCN achieved an MAE and MSE of 4.37 and 6.45, respectively, on the BeanCount-1500 dataset (see Table 4 for details). In the model performance generalization experiment on the Renshou2021 dataset, SPCN’s MAE and RMSE were 4.43 and 6.18, respectively. Compared with PodNet [3], the RMSE of SPCN was lower by 1.47, demonstrating its significant generalization potential in different scenarios.

Both the attention mechanism and the HDC strategy are incorporated in SPCN. The introduction of the attention mechanism enables the model to dynamically focus on important areas of the image, improving the efficiency in feature extraction. The HDC strategy increases the receptive field while maintaining computational efficiency by using convolution kernels with different expansion rates. This increase in the receptive field is particularly important for processing soybean plant images with complex backgrounds and diverse features, as it helps the model distinguish between different soybean pods more effectively. Additionally, SPCN uses the UOT loss function to supervise the generation of density maps. This function efficiently evaluates the minimum transmission cost between the target distribution and the predicted distribution, resulting in more accurate and refined density maps.

To evaluate algorithm performance, we constructed the BeanCount-1500 dataset. The dataset includes a total of 24,684 accurately annotated images of 1500 soybean plants covering plants with different morphologies and different growth environments with various weather conditions and light intensities. However, in complex situations, when pods are occluded by leaves and neighboring pods, the counting performance of SPCN is decreased to a certain extent. With only a customer-level NVIDIA GeForce RTX GPU
Agriculture 2024, 14, 1347

(Santa Clara, CA, USA), an image can be processed within 600 ms on average. We have also implemented SPCN in the WeChat Mini Program of the School of Agriculture, South China Agricultural University, for easy soybean plant phenotyping.

Therefore, there is still room for improvement as follows: 1. Consider introducing advanced data augmentation methods to cope with the challenges of complex scenarios; 2. Further improve the loss function to generate finer and more dispersed density maps, retaining the model’s excellent counting performance in high-density scenes while providing precise coordinate information of soybean pods more effectively; 3. The density map-based counting methods hold great promise for pod counting applications, especially for dense counting tasks. Future research could explore ways to simplify density map-based counting models, perhaps by integrating techniques such as knowledge distillation, to develop lighter models without compromising counting accuracy. This adjustment is designed to better meet the needs of practical agricultural applications. Through these efforts, new research directions are expected to advance pod counting technology and bring higher practical application value to the agricultural sector.

5. Conclusions

Soybean pod counting, a pivotal technique in precision agriculture, holds considerable practical significance yet encounters several challenges, including mutual occlusion and the absence of dedicated large datasets. To tackle these challenges, this study introduces an innovative approach, the density map based SPCN model, which integrates the HDC strategy as well as CBAM for feature extraction and UOT loss function for supervising density map generation to achieve highly accurate soybean pod counting. Additionally, we curated BeanCount-1500, a comprehensive dataset dedicated to soybean pod counting encompassing images of different soybean varieties with diverse backgrounds, lighting conditions, and varying numbers of plants, thereby providing a valuable resource for research in this domain.

Through extensive experiments on the BeanCount-1500 dataset, the SPCN model showcases superior counting performance over existing mainstream counting methods, achieving an MAE and MSE of 4.37 and 6.45, respectively. Detailed ablation experiments further affirm the effectiveness of the HDC strategy in soybean pod counting. Furthermore, the SPCN model trained on BeanCount-1500 achieved a high counting accuracy when tested on the Renshou2021 dataset, demonstrating its excellent generalization capability, achieving an MAE and RMSE of 4.43 and 6.18, respectively. This indirectly underscores the tremendous diversity covered by the BeanCount-1500 dataset, validating its valuable potential for future model development.

We anticipate that this study will make significant contributions to the advancement of precision agriculture and the development of smart counting systems. Furthermore, we aim to extend the BeanCount-1500 dataset in future research endeavors and to explore the feasibility of deploying the model on lightweight mobile devices. By doing so, we aim to preserve the benefits of density map-based counting methods in addressing dense scene counting while satisfying the demand for real-time counting in agricultural fields. Ultimately, this initiative will facilitate the seamless translation of this technology into practical applications, thereby enhancing efficiency and productivity in agricultural practices.

Author Contributions: Conceptualization, X.L., Y.Z. (Yitao Zhuang) and Y.G.; Data curation, J.L. and D.L.; Formal analysis, Y.Z. (Yue Zhang); Funding acquisition, X.L.; Investigation, X.L., D.L. and Y.G.; Methodology, X.L., Y.Z. (Yitao Zhuang), J.L., Y.Z. (Yue Zhang), Z.W. and Y.G.; Project administration, X.L. and Y.G.; Resources, J.L. and Y.G.; Software, Y.Z. (Yitao Zhuang); Supervision, X.L.; Validation, Y.Z. (Yue Zhang), Z.W., J.Z. and Y.G.; Writing—original draft, Y.Z. (Yitao Zhuang) and J.Z.; Writing—review and editing, Y.Z. (Yitao Zhuang), Z.W. and Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the Key Research and Development Program of Guangzhou (No. 2024B031J1358).
Data Availability Statement: The original findings detailed in this study are contained within the article. For additional inquiries, please contact the corresponding authors directly. We will open source the code at https://github.com/johnhamtom/soybean_counting_SPCN (accessed on 10 August 2024).

Acknowledgments: Thanks to everyone who participated in photographing and labeling soybean plants.

Conflicts of Interest: The authors declare no conflicts of interest.

References

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.