


Article

Seeding Crop Detection Framework Using Prototypical Network Method in UAV Images

Di Zhang ¹, Feng Pan ^{1,2,*}, Qi Diao ¹, Xiaoxue Feng ¹, Weixing Li ¹  and Jiacheng Wang ¹

¹ School of Automation, Beijing Institute of Technology, Beijing 100081, China; zhangdi.359@163.com (D.Z.); 3120185444@bit.edu.cn (Q.D.); fengxiaoxue@bit.edu.cn (X.F.); liweixing@bit.edu.cn (W.L.); 3220200796@bit.edu.cn (J.W.)

² Kunming-BIT Industry Technology Research Institute Inc., Kunming 650106, China

* Correspondence: panfeng@bit.edu.cn

Abstract: With the development of unmanned aerial vehicle (UAV), obtaining high-resolution aerial images has become easier. Identifying and locating specific crops from aerial images is a valuable task. The location and quantity of crops are important for agricultural insurance businesses. In this paper, the problem of locating chili seedling crops in large-field UAV images is processed. Two problems are encountered in the location process: a small number of samples and objects in UAV images are similar on a small scale, which increases the location difficulty. A detection framework based on a prototypical network to detect crops in UAV aerial images is proposed. In particular, a method of subcategory slicing is applied to solve the problem, in which objects in aerial images have similarities at a smaller scale. The detection framework is divided into two parts: training and detection. In the training process, crop images are sliced into subcategories, and then these subcategory patch images and background category images are used to train the prototype network. In the detection process, a simple linear iterative clustering superpixel segmentation method is used to generate candidate regions in the UAV image. The location method uses a prototypical network to recognize nine patch images extracted simultaneously. To train and evaluate the proposed method, we construct an evaluation dataset by collecting the images of chilies in a seedling stage by an UAV. We achieve a location accuracy of 96.46%. This study proposes a seedling crop detection framework based on few-shot learning that does not require the use of labeled boxes. It reduces the workload of manual annotation and meets the location needs of seedling crops.

Keywords: chili detection; prototypical network; small-scale similarity problem; unmanned aerial vehicle images



Citation: Zhang, D.; Pan, F.; Diao, Q.; Feng, X.; Li, W.; Wang, J. Seeding Crop Detection Framework Using Prototypical Network Method in UAV Images. *Agriculture* **2022**, *12*, 26. <https://doi.org/10.3390/agriculture12010026>

Academic Editor: Koki Homma

Received: 4 November 2021

Accepted: 23 December 2021

Published: 27 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Object detection from optical remote sensing images plays a vital role in image interpretation. It involves locating and recognizing objects of interest from a given image. Generally, there are differences between the detection of specific object classes such as cars [1,2], ships [3,4], aircraft [5], and general object class detection [6]. With the development of UAV technology, it has become easier for people to obtain high-resolution aerial images, which play an important role in agricultural applications [7–11]. In modern agriculture, crop and field information can be easily obtained through aerial images, which is useful for crop growth monitoring, planting density research, and agricultural insurance business.

Aerial images have unique characteristics. The viewing angle is basically a high-altitude top view, which is quite different from natural images, and the views of the same object are different. Therefore, a model trained on a natural scene image dataset may give poor results when used for aerial images. However, only a few aerial image datasets are available for crop detection, and they do not contain the images of the required crop species. Therefore, studying the recognition of specific crops from aerial images is still meaningful.

Aerial image crop detection mainly involves two methods: object-based and pixel-based. Mafanya et al. evaluated pixel-based and object-based classification techniques [12]. Object-based detection in traditional methods has an outstanding performance [13]. However, in deep learning methods, pixel-based mask region-based convolutional neural networks (R-CNN) have also achieved good results in object detection tasks [14]. Object-based detection methods generally involve two processes: candidate region extraction and classification. There are several methods for extracting candidate regions using sliding windows. Object detection based on deep learning can be divided into two-stage frameworks (such as R-CNN series) and single-stage frameworks (such as YOLO series). The candidate region extraction in R-CNN uses a selective search method, which is an earlier version of a superpixel segmentation method. In this study's detection task, we use the simple linear iterative clustering (SLIC) method [15] for the candidate region extraction and a prototypical network for the classification.

In several studies, superpixel segmentation images are used to generate image patches to create datasets [16,17]; they are rarely used directly for object detection. Some studies have been conducted on crop detection from UAV aerial images. For example, Malek et al. [18] detected palm trees in an image. They used scale invariant feature transform (SIFT) to extract a set of key points on a given UAV image, and then an extreme learning machine (ELM) to analyze them. The ELM classifier marks each detected palm tree with several key points. Ha et al. [19] proposed a computerized system for detecting Fusarium wilt in radish. It divides the entire radish field into three unique regions using a softmax classifier and K-means clustering, and then uses CNN to further classify healthy radish and radish with Fusarium wilt. Wang et al. [20] used the HOG algorithm to extract palm tree texture features, and then a support vector machine (SVM) to perform a classification for detecting a single oil palm tree in UAV images. For tobacco detection, a watershed segmentation method was used by Fan et al. [21] for extracting candidate regions, and then a convolutional neural network was used in the classification stage. Many studies have focused on weeds [22–26], trees [27], and grapes [28]. Donmez et al. [29] detected and counted orchard citrus trees from a UAV image. They used a comprehensive approach by combining a pixel-based CCL algorithm with morphological image operations to detect each plant individually using rectangles. Lin et al. [30] used a deep learning model to detect and count cotton plants at a seedling stage using UAV images. The dimensions of the training images were 300×400 pixels. These models were trained using two datasets containing 400 and 900 images, respectively. Their results showed that more training images are required when applying object detection models to images with different dimensions from the training datasets. Tetila et al. [31] used UAV images to identify soybean leaf diseases. They used a superpixel method to segment the image and then manually selected samples to create a dataset. Subsequently, the dataset was classified and evaluated, and object detection was not performed on the image. There is no unified standard detection framework for the detection of seedling crops from large-field optical UAV images. This study attempts to perform a detection task with one large-field UAV image as the dataset.

The location and quantity of crops are important for agricultural insurance businesses. In this study, a UAV image-based crop detection framework is proposed. To evaluate the proposed framework, the authors collected UAV images of seedling chilies. A few data are used for crop detection, and a label box is not required. This alleviates the intensive task of labeling. A solution is proposed for the small-scale similarity problem in aerial crop images.

Objects in aerial images are generally small and consist of only a few pixels. If the crops are densely planted, the sample images in the dataset may contain other interfering information. During a detection, a patch image extracted near the target may be difficult to recognize, which may make it difficult to accurately locate the crop. The patch image that may be extracted near the target during the detection is shown in Figure 1. On a smaller scale, different objects show similar characteristics, leading to classification errors and reduced network performance. This is known as the small-scale similarity problem in

the literature [32]. The candidate regional patch images generated near the object in the image have similar characteristics, which increases the classification and location difficulty.

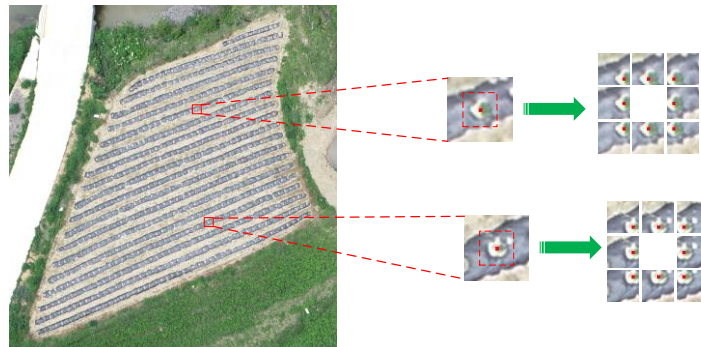


Figure 1. Patch images extracted around the object during the detection process.

Inspired by the similarity of the patch images extracted around an object during the detection process, the object sample images were further divided into several subcategories for a better location detection. Regarding the study of subcategories in image recognition, Dong et al. [33] proposed an ambiguity-guided subcategory mining approach to explore the intrinsic subcategory structure in each category. This method combines intra- and inter-class ambiguities to effectively realize subcategory mining. Chen et al. [34] proposed a subcategory-aware network S-CNN to design an instance-sharing maximum boundary clustering algorithm and a subcategory-aware convolutional network, which effectively alleviated the problem of large intra-class differences in object detection. As we know, detection methods based on deep learning need to label a large number of samples. At the same time, studies have also explored and made some progress with few-shot learning. A high accuracy rate has been achieved using the Omniglot dataset [35]. The recognition accuracy on datasets such as miniImageNet [36] is also constantly improving. In this study, a prototypical network for few-shot learning is used to establish a classification model for chili recognition.

Innovation

First, the proposed detection framework is innovative in that it uses the SLIC method to segment an image, superpixel center points to generate the candidate region, and then a prototypical network for classification and location detection. Currently, it seems that no published literature has introduced this detection method. Second, the object sample images are further divided into several subcategories for a better location accuracy. Meanwhile, a prototypical network is introduced, which can be used to complete a detection task with a small number of samples. Finally, the method of simultaneous detection using nine patch images is also novel.

In this study, a method to detect seedling chilies from UAV aerial images is proposed. The chili images are classified into fine subcategories, and a prototypical network is used to establish a classification model. In the detection process, the superpixel method is used to segment a UAV image, following which the candidate regions are generated. A search is performed around the candidate regions, and the nine patch images are concurrently extracted for recognition to locate the chilies; the effectiveness of this method is experimentally verified. The rest of the paper is organized as follows. The second section introduces the materials and methods used in this study. The third section reports the study results. The fourth section includes a discussion of the results and the fifth section concludes the study.

2. Materials and Methods

2.1. Overall Recognition Framework

The detection of chilies from aerial UAV images involves two stages: creation of a dataset to train a prototypical network and detection. The images are shown in Figure 2, and we aim to locate the seedling chilies in these images.

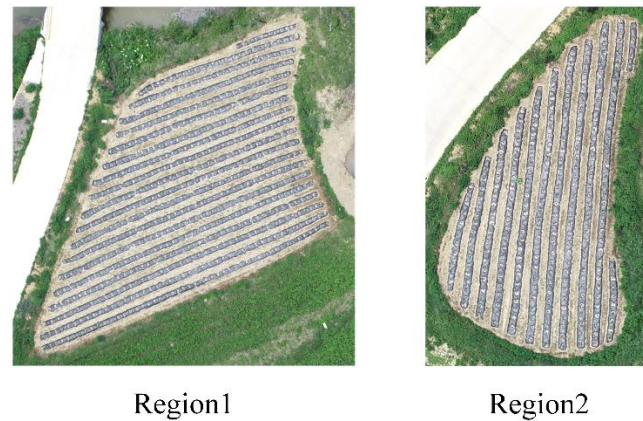


Figure 2. The regions containing seedling chilies to be detected in the UAV aerial images.

The processing includes the creation of a dataset to train the prototypical network. The input image is segmented by superpixels to generate candidate regions; following which a prototypical network is used to classify the generated patch images to remove the center point of the background class; the prototypical network is used to classify the generated patch image of the candidate regions and calculate the similarity distance with the prototype, and the crop location process is then completed. The main parts of the system are shown in Figure 3.

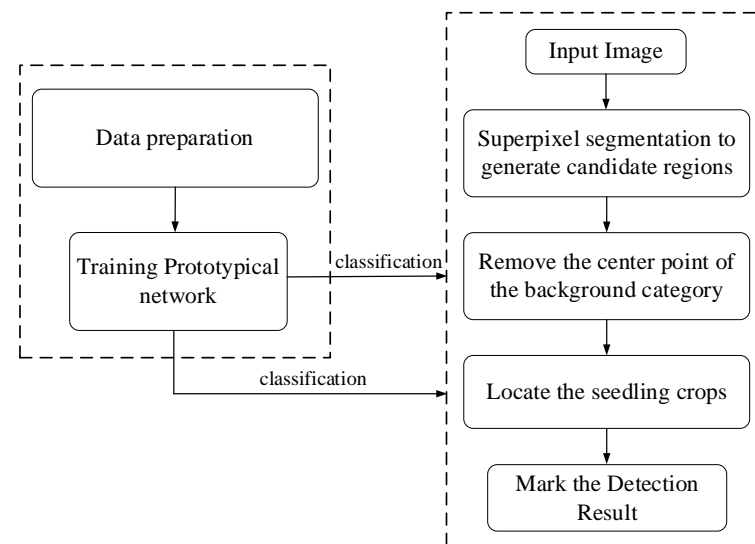


Figure 3. Processing architecture.

2.1.1. Data Preparation

The images used in this study to detect chilies at the seedling stage were taken in Yunnan by a UAV. Region 3 in the image is selected to create a dataset for the training model, and Regions 1 and 2 are the images to be detected, which are manually marked as the ground truth. The detection effect is evaluated by comparing the results predicted by the model with the ground truth. The image sizes of Regions 1, 2, and 3 are 1962×2061 , 963×1518 , and 3423×1397 , respectively. For creating the dataset, Python was used to

develop a simple and automatic sample cutting software with a GUI interface. The user only needs to click on the input image. When labeling chili samples, the center point of the chili object in the image is clicked, and a 30×30 patch image will be generated as the main category of the chili sample, as shown in the red dashed box in Figure 4a. The eight chili subcategory patch image blocks around it are automatically cut after offsetting the fixed pixel value with the marked point as the center. The purpose of the red marked point is to show the condition of the fine-grained subcategory patch and coarse-grained main category patch in Figure 4a. There are no red marked points on the sample images in the training set. The sample image size of the training set is 30×30 . When the sample image is input to the network, its size becomes 84×84 following a resize. The method proposed in this study does not use bounding boxes, which saves time and effort in the labeling stage of the sampling process. Figure 4b shows three non-chili background categories, and the background patch images are obtained by clicking the corresponding positions on the Region 3 image, and then automatically cutting the background samples. As for the nine categories related to chilies, 500 sample patch images were generated for each. As for the three categories not related to chilies, 100 sample patch images were generated for each of them.

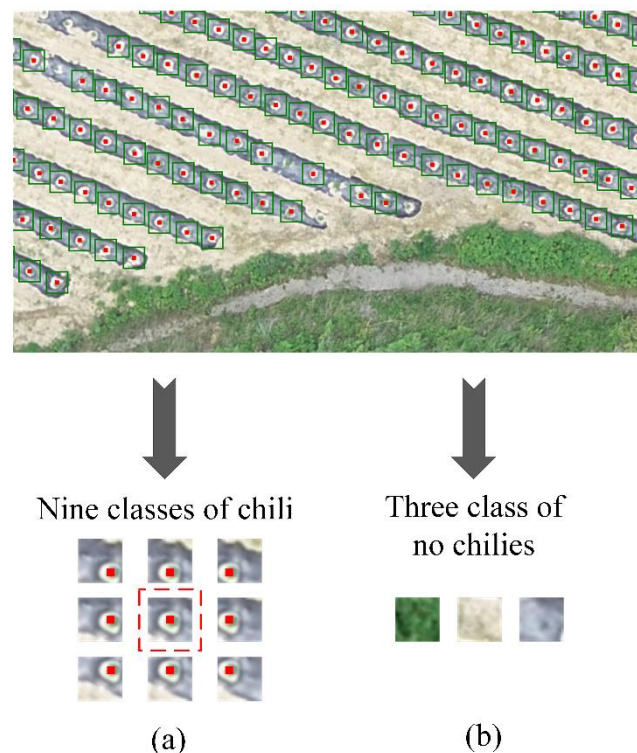


Figure 4. Schematic diagram of dataset be produced. (a) Main chili category and its eight chili subcategories are sampled. (b) Patch images of background categories are sampled for the dataset.

2.1.2. Detection Flow

The overall structure of the UAV aerial image chili recognition system proposed in this study is shown in Figure 5. It primarily includes the training and detection processes. In the training process, aerial images are used to create sample patch images of the 12 categories to train the prototypical network and classify them. Among them, nine categories correspond to chilies and the other three categories correspond to the background. From these 12 categories in the training data set, 40 samples are taken from each category as the prototype of the prototypical network for the detection process.

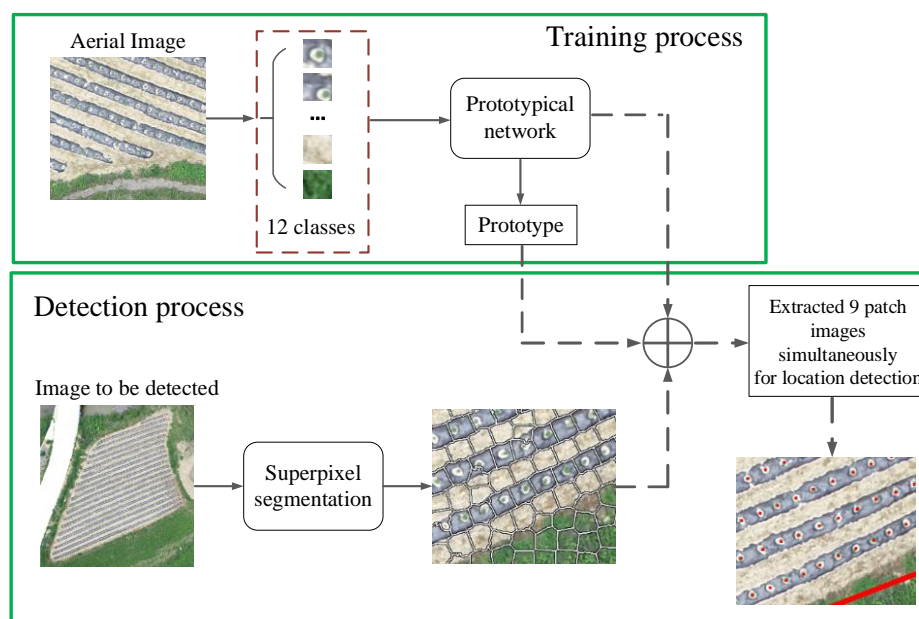


Figure 5. Overall architecture.

In the detection process, a supersixel segmentation is used to process the aerial images, and the center point of each supersixel as the center is used to generate the candidate regions for detection. Around the candidate regions, nine patch images are generated simultaneously, and then input to the prototypical network, following which the metric value is calculated using the prototype. Then, the position of the chili object center points in the aerial is obtained. In the next section, various algorithms pertaining to this system are introduced in detail.

2.2. Method for Chili Detection from UAV Image

2.2.1. Classification of Chilies with Prototypical Network

Prototypical networks [37] are used as a model for few-shot learning. Humans are particularly good at learning from examples. To learn a desired task from limited supervised information, the problem of few-shot learning is proposed. The classifier is trained on a base class dataset with a large number of examples to learn to recognize new classes that are not in the base class. The new class provides only a few labeled examples. For example, in the miniImageNet dataset, there are 100 classes, each with 600 image examples. We follow the split introduced in [38], which has 64 classes for training, 16 classes for validation, and the remaining 20 classes for testing. The few-shot classification is based on a N -way k -shot classification task, where N classes of images are randomly selected in the new classes, and k labeled examples in each class are provided as a support set. Fifteen images can be selected as query images from each class, and then the trained model can be used to predict the classes of these query images. References [36,37] used an episode training method to complete the few-shot task, making the training environment more similar to the test environment, which is conducive to improving the generalization performance of the test phase. The purpose of classifying chilies in this study is to locate them in UAV images using a prototypical network in the classification stage of the detection task. The episode-based approach is used in the training stage and not in the testing and location detection stages.

According to the model classification, a prototypical network is a type of embedded learning [39]. The embedded learning method transforms a sample into an embedded space through a prototypical network. The embedded function in the prototypical network is represented by $f_{\phi} : R^D \rightarrow R^M$, where ϕ is a learnable parameter. In the N -way K -shot task, N represents the number of categories in an episode, and the number of samples in each category in the support set is K . The representation of each category in the embedded space can become a prototype, and each prototype is the average value of the embedded support

vectors belonging to its own category. The prototype of the k -th category is expressed in Equation (1) as

$$c_k = \frac{1}{K} \sum_{i=1}^K f_\phi(x_i), k \in \{1, \dots, N\} \tag{1}$$

Given a distance function $d : R^M \times R^M \rightarrow [0, +\infty)$, a class distribution is developed through the prototypical network for a query point x based on a softmax over distances to the prototypes in the embedded space [35]. The predicted probability of true class k is expressed by Equation (2) as

$$p_\phi(y = k|x) = \frac{\exp(-d(f_\phi(x), c_k))}{\sum_{k'} \exp(-d(f_\phi(x), c_{k'}))} \tag{2}$$

The distance $d(\cdot, \cdot)$ is a Euclidean distance in this study. The learning stage minimizes the negative log-probability $J(\phi)$ of the true class k via the Adam optimizer. It can be referred to as the compute process of the loss function $J(\phi)$ for a training episode in the reference [37] as follows:

$$J(\phi) = -\log p_\phi(y = k|x) = d(f_\phi(x), c_k) + \log \sum_{k'} \exp(-d(f_\phi(x), c_{k'})) \tag{3}$$

The classification process measures the distance between the prototype and query images in the embedded space. The embedding network of several few-shot learning models uses four convolution modules for stacking; each convolution module contains a 3×3 convolution with 64 filters, a batch normalization, a ReLU nonlinear layer, and a 2×2 max-pooling layer.

For the location detection, the chilies are divided into nine categories in this study; there are 12 categories in total, which includes three background categories. A prototypical network is trained using an episode-based approach to distinguish between the 12 categories. Because the prototypical network trained in this study does not need to recognize new categories, the episode-based approach is no longer used for testing. The categories and images are randomly selected when the episode-based approach is used for testing. In this case, the testing for each image in the test set is generally not completed. In this study, when using the trained prototypical network for classification testing, each image in the test set is used as a query image for testing. The test image obtained using the prototypical network is shown in Figure 6.

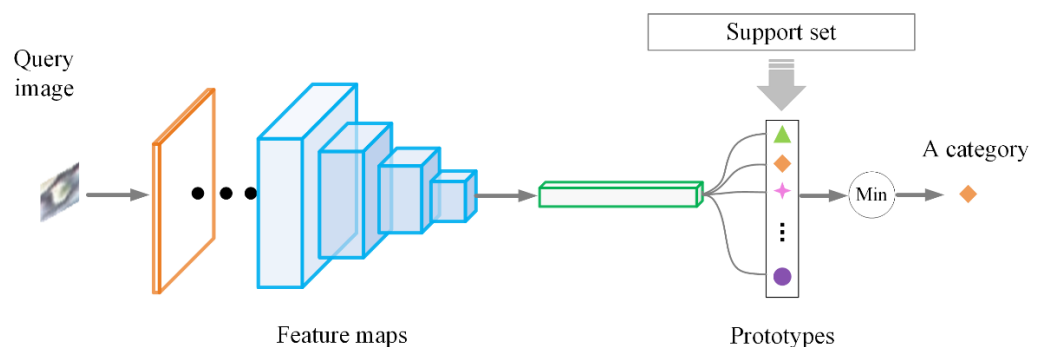


Figure 6. Block diagram of the prototypical network for chili classification. The query image is taken from the test set. It generates a 1×1600 vector after being embedded in the network.

The support set generates prototypes for each category through an embedded network, which are then used for the classification. The embedding vector of the query image is compared with each prototype embedding vector in the support set using Euclidean distance, and the closest prototype category is selected as the predicted category of the query image.

In the chili classification task, because the number of all categories is small, they are treated as a support set during training and testing. During testing, the images in the training set are used as a prototype, and all images in the test set are traversed for the prediction.

2.2.2. Superpixel Segmentation to Generate Candidate Regions

(1) Superpixel segmentation to generate central points.

The superpixel concept was proposed by Ren et al. [40]; it refers to irregular pixel blocks composed of adjacent pixels with similar colors or other low-level attributes. There are several methods for generating superpixels. In this study, the SLIC method [15] is used to perform an unsupervised segmentation of the image to be detected, as shown in Figure 7. There is only one parameter k in this algorithm, which represents the number of superpixels to be segmented. For color images, it will be better to use the CIELAB color space.

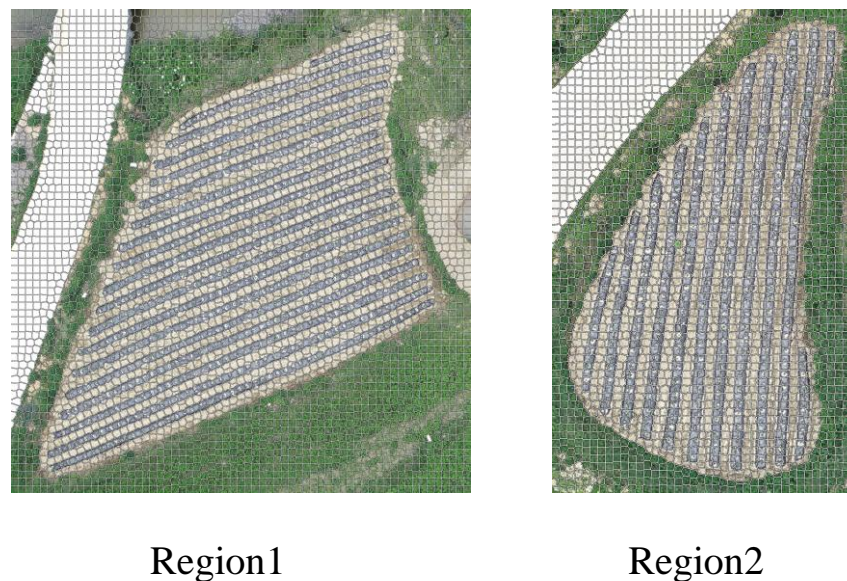


Figure 7. Superpixel segmentation of the images to be detected using the SLIC method.

For the case where the number of input image pixels is N and the number of superpixels expected to be generated is k , the height and width of a superpixel are both $S = \sqrt{N/k}$ at the beginning, and then the clusters are in the 2S range. When using the CIELAB color space, clustering is performed in the labxy space, which combines color distance and spatial distance into a metric shown in Equation (4):

$$\begin{aligned}
 d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}, \\
 d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \\
 D' &= \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2}.
 \end{aligned} \tag{4}$$

Through a superpixel segmentation of the image to be detected, each superpixel image block will have a cluster center point; these center points are saved for the subsequent operations.

(2) Superpixel central-point selection

After the image to be detected is subjected to superpixel segmentation, several superpixels are generated. To avoid detecting a large amount of useless background information, the detection area is marked. The marking process is quite simple. Using the software for marking samples described in the text, one needs to only mark a few points and connect them sequentially. The area enclosed in the middle is to be detected, as shown in Figure 8b. Then, the points inside the marked polygon are filtered out. The ray method can be used to

determine whether a point is inside the polygon. The basic concept is to draw a ray from a point that needs to be judged, and then calculate the number of intersections between that ray and the polygon. If the number of intersections is odd, the point is located inside the polygon; if the number of intersection points is even, the point is outside the polygon. As shown in Figure 8d, the point is located outside the polygon. Following this selection, the center point of the superpixel in the marker box is shown in Figure 8c.

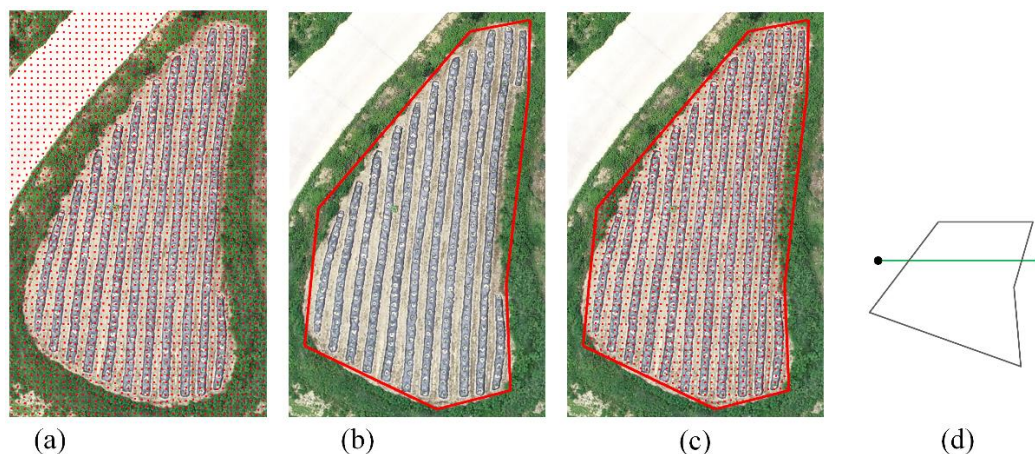


Figure 8. The center points of the superpixel in the marker box are selected. (a) All center points of Region2 after superpixel segmentation. (b) Marked area that needs to be located in Region2 by a polygon box. (c) Center points of the superpixel in the polygon box selected by the ray method. (d) Using the ray method to determine whether the point is inside the polygon.

(3) Region proposal

The patch image is extracted from the image to be detected to detect chilies. Then, the extracted patch images are classified. The superpixel patch image produced by superpixel segmentation is irregular and cannot be directly sent to the classifier for a classification. Moreover, it is difficult to use the current unsupervised superpixel segmentation method to appropriately segment the chili image. In this study, the cluster center point of superpixel segmentation is used as the center to generate a patch image with the same sample size. When proposing the use of these patch images as the proposed regions for the subsequent operations, it is obvious that the generated patch image candidate regions are affected by superpixel segmentation. After marking the polygon frame on the image to be detected, it is only necessary to perform the subsequent operations for the superpixel center point in the polygon frame.

2.2.3. Removal of the Background Category's Center Point

After using superpixels to segment the image to be detected, many superpixel patch images that are obviously background types will be obtained. Based on Figure 8c, the candidate region patch images are generated by extracting the superpixel cluster center points, which will be subsequently recognized.

The patch images generated from these center points in the marked box are sequentially input into the prototypical network as query images. The support set in the prototypical network uses the images in the training set, and all 12 categories are used as support sets. Then the prototypes in the prototypical network are generated. In the detection phase, the center points identified as the background category, such as the one in Figure 4b, are removed. The center points of the superpixel after removing the background category in Region2 are shown in Figure 9. Although these center points do not include background categories, chilies are still not located.



Figure 9. Superpixel segmentation center points of Region2 are displayed after removing the background categories.

2.2.4. Location Method

The aforementioned classification and candidate region extraction are for object-locating services. Generally, objects in an image classification experiment are often centered. However, the generated patch images of the candidate regions will contain images whose objects are not at the center or those with incomplete objects. On a smaller scale, this issue may have a greater impact, and these patch images could be recognized as background types. The generated patch images of these candidate regions increase the difficulty of classification and location. In this study, the proposed method for classifying the patch images extracted around an object can better help with the location of chilies. Moreover, based on the method of classifying the patch images around chilies, patch images around the candidate region are extracted at the same time for recognition, and then the location detection is performed after a comprehensive calculation.

After removing the background categories, the superpixel segmentation center points near the chilies are obtained, as shown in Figure 9. Suppose a certain point is taken as the center to generate a 30×30 patch image. Then, eight patch images are taken at equal distances around it, and then these nine patch images are input into the trained prototypical network in a fixed order, as shown in Figure 10. Each patch image generates an embedding vector after passing through the embedding network, and each embedding vector only measures the Euclidean distance with the prototype category at the corresponding location instead of comparing the embedding vector with each prototype category. These nine metric values are then summed for the location process.

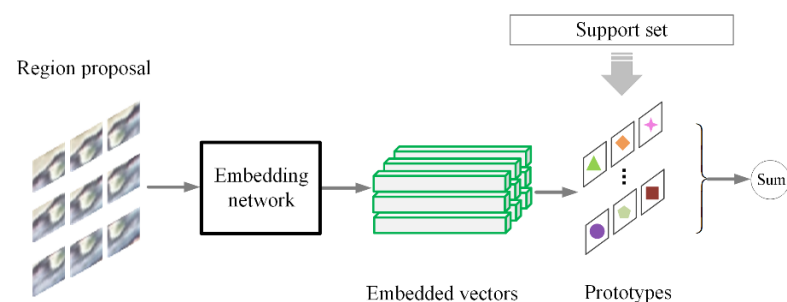


Figure 10. Prototypical network is used to simultaneously recognize nine extracted patch images.

Taking a certain superpixel segmentation center point as the center, a 30×30 patch image is selected. This patch image is set as Category 1, and its left patch image is set as category 2. Then, the patch image is taken in a clockwise direction, and finally in the lower left corner as Category 9. These nine patch images are used as the input unit set $Q = \{q_1, \dots, q_N\}$, where $q_i \in R^D$ is a D -dimensional input vector and N is equal to 9. The input vector of each patch image is mapped to the embedding space through the embedding network, and the embedded function is represented by $f_\theta : R^D \rightarrow R^M$; the M -dimensional space is the transformed embedded vector space. In the embedding space, the prototype vector generated by the support set is recorded as $P = \{p_1, \dots, p_N\}$. The distance between each input patch image and the prototype in the embedding space is shown in Equation (5). Then, these values are added as shown in Equation (6). In the detection task, the extracted candidate region patch image is fed to the prototypical network for a classification based on its distance from each prototype in the embedding space. Extracting nine patch images sequentially at the same time is similar to increasing the context. The whole with this increasing context is similar to the whole composed of nine category prototypes when the distance sum calculated by Equation (3) is the smallest. Then, we can consider the location of the extracted center candidate region to be the object's location.

$$d_i = \|f_\theta(q_i) - p_i\|_2 \quad (5)$$

$$s = \sum_{j=1}^N d_j \quad (6)$$

The chilies are located based on the superpixel segmentation center point after removing the background categories. Suppose these center points are marked as $C = \{(x_1, y_1), \dots, (x_K, y_K)\}$, where K is the number of center points. Each center point coordinate is searched and located in the surrounding area $\eta \times \eta$, where η is the search range parameter. L points are generated in this area based on interval σ and marked as (x_k^l, y_k^l) , and nine patch images are generated with each point as the center. As a set of candidate regions, they are marked as $Q_k^l = \{q_1, \dots, q_N\}$, where N is equal to 9. These will be fed to the prototypical network as an input unit set. Then, the Euclidean distance between each patch image q_n in the input unit set and the prototype vector p_n of the corresponding position category in the embedding space is calculated, as shown in Step 7 of Algorithm 1. Then, we add them, as in the 8th step of Algorithm 1, and the sum of the measurement values of the input unit set of the l th point is s_k^l . The value of point L is recorded as (x_k^l, y_k^l) . Then, the minimum value of S_k and the coordinate (x_k^l, y_k^l) of the corresponding l th point are considered as the output location coordinate in this $\eta \times \eta$ area, that is, the output coordinates of the k th superpixel segmentation center point after executing the location detection algorithm, as shown in the 10th step of Algorithm 1.

After obtaining the above output coordinate points, the points closer to each other are merged. In other words, for a group of points with similar distances, the average values of the horizontal and vertical coordinates of each point are used as the location coordinates after merging them.

Algorithm 1 Location recognition algorithm based on superpixel center point**input:** The center point coordinate of superpixel segmentation $C = \{(x_1, y_1), \dots, (x_K, y_K)\}$ **output:** Coordinates after location**Initialization:** η, σ

```

1: for k = 1, 2, ..., K do
2:   Obtain the region of  $\eta \times \eta$  //The upper left and bottom right coordinates of the
   region are
                                      $(x_k - \eta, y_k - \eta)$  and  $(x_k + \eta, y_k + \eta)$ , respectively.
3:   Sampling  $L$  points from this region with  $\sigma$  as an interval.
4:   for  $l = 1, 2, \dots, L$  do
5:      $(x_k^l, y_k^l) \rightarrow Q_k^l = \{q_1, \dots, q_N\}$  //Generate the input unit set of the
                                               candidate region image
patch.
6:     Feed the input unit set into the prototypical network.
7:     Calculate the Euclidean distance according to Equation (2).
8:     Obtain  $s_k^l$  by adding these metric values according to Equation (3)
9:   end for
    $S_k = \{s_k^1, \dots, s_k^L\}$  //Record the value of these  $L$  points.
10:   $\min(S_k) \rightarrow (x_k^l, y_k^l)$  //Output location coordinates of
                                     the  $k$ -th superpixel center point.
11: end for

```

3. Results

This section presents the test results. First, the experimental results of the prototypical network used in the classification process are described, which is followed by the results and an analysis of the chili detection and location process. Finally, the impact of the number of samples on the network model is explored. We used the PyTorch deep learning framework and NVIDIA GeForce RTX 2080 Ti graphics card for the experiments.

3.1. Chili Classification Experiment Results

In the proposed chilies detection method, training the prototypical network for classification is required. Region3 is used as the training set to create samples. Moreover, each class has 500 samples, and the classification experiments are performed on the datasets from Region1 and Region2. The prototypical network is trained in an episode-based manner with the training set. A fixed number of shot and query images in each episode are taken randomly from each class. A shot image is used to generate the prototype vector, and the model's loss-decline curve with different shot numbers is shown in Figure 11.

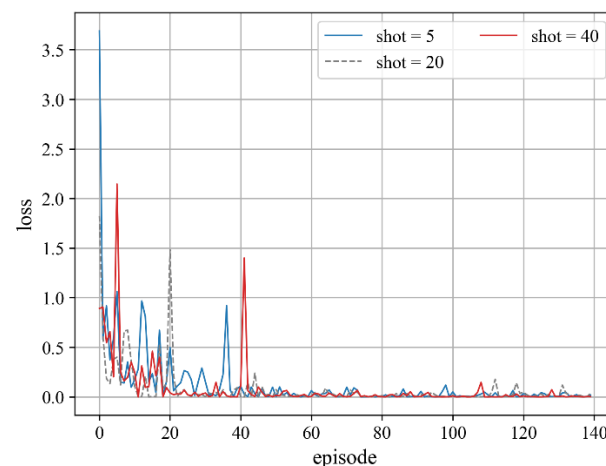


Figure 11. Loss-decline curve for different shots.

After obtaining the trained prototypical network, the test is performed on the datasets from Region1 and Region2. Standard few-shot classification experiments generally use the same training and test environments, that is, episodes are used to test the network performance during testing. If the chili classification experiment is also tested in this manner, the traversal of all images in the test set cannot be guaranteed. Therefore, the episode-based approach is not used in this study's classification experiment, but all images in the test set are tested. When the prototypical network is used for testing, it is necessary to obtain images for generating the prototype. In this study, images in the training set are randomly selected to generate prototypes. The prototypical network is trained with different numbers of samples, and the accuracies of the test set under different shots are shown in Table 1.

Table 1. Classification accuracies on chili dataset by the prototypical network.

Number of Samples	Test Data	ACC		
		Shot = 5	Shot = 20	Shot = 40
50	Region1	55.082%	88.996%	92.204%
	Region2	44.514%	80.486%	84.826%
100	Region1	89.730%	95.157%	92.054%
	Region2	77.604%	90.972%	82.535%
500	Region1	98.321%	98.172%	98.456%
	Region2	94.236%	96.528%	96.563%

3.2. Chili Detection Experiment Results

To evaluate the detection method proposed in this study, the images to be detected in Region1 and Region2 were manually annotated, and the detected chili coordinates were compared with the ground truth. This study uses producer's accuracy ($Pacc$), user's accuracy ($Uacc$), and average accuracy (Acc). $Pacc$ (Equation (7)) is the percentage of chilies correctly detected in all test results, and $Uacc$ (Equation (8)) is the percentage of chilies correctly detected in the total number of ground truths.

$$Pacc = \frac{TP}{TP + FN} = \frac{TP}{N} \quad (7)$$

$$Uacc = \frac{TP}{TP + FP} \quad (8)$$

$$Acc = \frac{Pacc + Uacc}{2} \quad (9)$$

where TP is true positive (that is, the number of chilies detected correctly), FP is false positive (that is, the number of objects falsely detected as chilies), and FN stands for false negatives (that is, the number of chilies missed). N represents the number of real chilies in a UAV image. In the test images from Region1 and Region2, if the Euclidean distance between the detected chili center point coordinates and the ground truth chili center point coordinates is less than or equal to 7 pixels, we consider the chili to have been correctly detected.

In addition, the relative error (Equation (10)) was used for the measurement. $N_p = TP + FP$ is the number of chilies detected.

$$Error = \frac{|N_p - N|}{N} \times 100\% \quad (10)$$

The detection results of Region1 and Region2 are shown in Table 2, and the average accuracies are 96.73% and 96.19%, respectively. High scores were obtained for $Pacc$ and $Uacc$. This shows the proposed algorithm can satisfactorily locate and count chilies in UAV

aerial images. The average relative error of the proposed algorithm is 4.32%, indicating that it performs well in yield estimation. The results of chili detection using the proposed algorithm are shown in Figure 12.

Table 2. Detection results of the proposed algorithm.

Area	<i>N</i>	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>Pacc</i> (%)	<i>Uacc</i> (%)	<i>Acc</i> (%)	<i>Error</i> (%)
Region1	662	625	6	37	94.41	99.05	96.73	4.68
Region2	278	262	5	16	94.24	98.13	96.19	3.96
Average	470	443.5	5.5	26.5	94.33	98.59	96.46	4.32

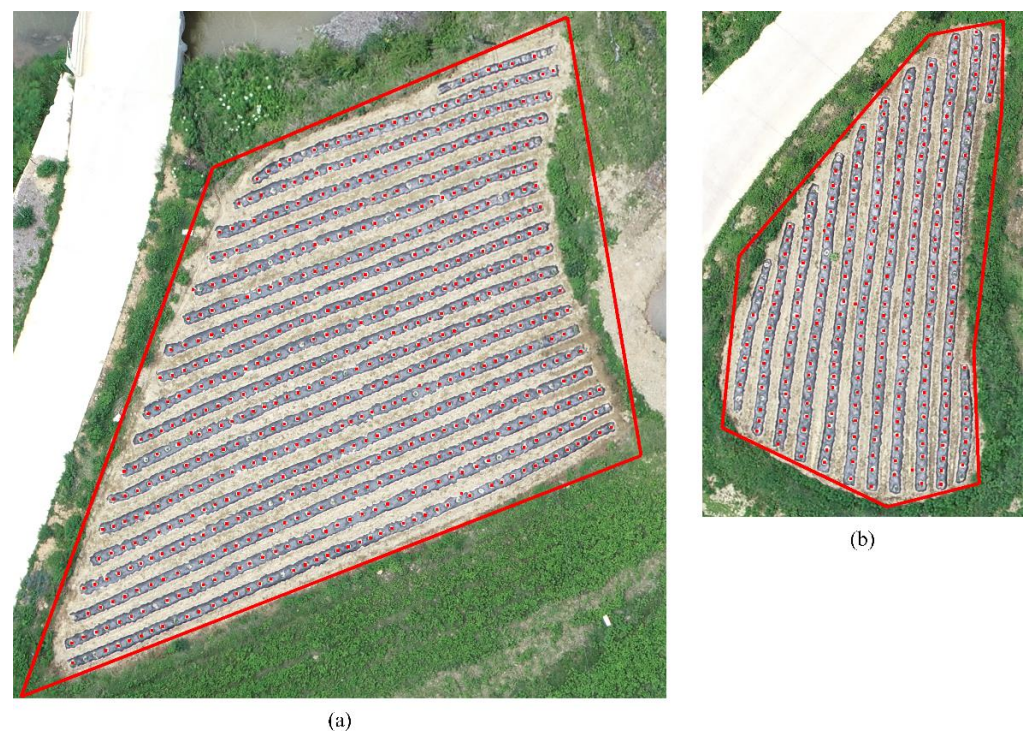


Figure 12. Results of chili detection in the two regions to be detected. The red block in the image is used to mark the detected chili crop center points. (a) Detection result of Region1. (b) Detection result of Region2.

In this study, a template-matching method is also used to detect chilies. The color template is selected from the chili classification training set. One hundred chili samples are randomly selected, and the average value of each channel of the color sample image is taken as a template for chili recognition. In the detection stage, the center point following the superpixel segmentation process is still used to extract the patch image in the 2S range around each center point, and the template-matching method is used for recognition. The center point of the patch image most similar to the template is taken as the location point of the chili. The average accuracies of Region1 and Region2 are 62.65% and 33.64%, respectively. Lin et al. studied the detection of seedling cotton crops in UAV images [40]. Their mean average precision and average recall were 79% and 73%, respectively, for the CenterNet model with 900 training images. Seedling cotton crops may vary significantly in scale.

4. Discussion

In this study, UAV images for seedling chilies were detected using the proposed framework. Region3 was used to create a data set, and the detection was performed on Region1 and Region2. The average *Pacc* and *Uacc* were 94.33% and 98.59%, respectively,

and the error was 4.32%. Good detection results were obtained, indicating that the proposed framework is effective and the detection task can be completed with a few samples.

When considering factors such as creation of a data set and complexity of detection methods in crop detection based on UAV images, there is rarely a generally accepted and satisfactory solution, especially when it comes to specific detection problems. To detect tobacco, the watershed segmentation method was used by Fan et al. [21] to extract candidate regions, and then a convolutional neural network was used in the classification stage. The framework proposed in this study uses superpixel segmentation to process the image for extracting candidate regions and then process it again. In the classification stage, a prototypical network was used in few-shot learning. Furthermore, the proposed method of subcategory slicing, when combined with the use of a prototypical network, can solve classification and location detection problems in the absence of pre-trained networks to some extent.

The superpixel segmentation method was used to identify soybean leaf disease from UAV images by Tetila et al. [31]. The purpose of using superpixels to segment images is to create a dataset rather than perform object detection on the image. We use the SLIC method [15] in superpixel segmentation to segment an image for candidate region extraction and then use the prototypical network for a classification to help locate chilies. Lin et al. [30] detected cotton plants from UAV images using deep learning models. General object detection networks based on deep learning are limited in terms of the input image size. Our method has no limitations in terms of the input image size. The seedling cotton plants detected by Lin et al. [30] can be multiscale, but our method is not yet able to detect multiscale objects. For multiscale detection, the proposed method needs to be improved continuously.

Our detection method does not require the use of a label box when creating the training dataset when compared with general deep learning-based object detection. Moreover, a method based on deep learning usually requires several samples for training. Even if there is a fine-tuning method, the network needs to be pre-trained. Moreover, even when a fine-tuning method is used, the recognition result may be unsatisfactory if the object category to be recognized is significantly different from the categories in the pre-training dataset. As for the UAV images for seedling crops in this study, there are hardly any data sets of similar categories. In the proposed detection framework, the network does not require pre-training. The classification network uses a prototypical network for few-shot learning.

We believe the solution proposed in this study is suitable for dense seedling crop detection from large-field optical UAV images, which are characterized by a small-scale similarity problem [32]. The use of deep learning methods generally requires a number of samples for training. When generating training samples, it is necessary to label the box for a general object detection network. Usually, there are one or several objects in a sample image. However, it is not easy to use this framework to process UAV images of large-field dense seedling crops, especially when only a few images are collected. In the detection of seedling chilies in this study, although the objects appear to have clear edges and are easy to handle, their size in the image is relatively small, which will cause some problems. Because the sample size is small, the extracted patch images that deviate far from the object are also recognized as samples when extracting candidate regions for recognition. This is the small-scale similarity problem mentioned above, which can easily lead to inaccurate location detection. This is briefly described in Figure 1. Considering this scenario, we propose a subcategory slicing method. To verify this problem, we use Region3 for the sample set and manually mark the center point of the seedling chilies. The sample patch image size is 30×30 pixels. The position offset by 7 pixels around the manually labeled center point is used as the center point of the subcategory, and then these patch images are taken as the subcategory sample images. The experimental results show that these categories can be correctly classified by the trained prototypical network without pre-training in the case of such small differences in subcategories. Regarding the collection of samples of subcategories, the extreme case is offset by one pixel. However,

the workload of the experiment and the main task in this study is to verify whether the proposed detection framework is effective. When taking samples of the subcategories, the offset pixel is 7. The sample size is 30×30 pixels, and the offset pixel is approximately a quarter of the sample size. The final seeding chili classification result is ideal.

In the detection stage, the author is inspired by the concept of context in image recognition. The eight fixed positions around each object may have a certain connection with the object itself. During the detection, after extracting the candidate region image, the patch images are also extracted at a position offset by 7 pixels around it. It is then combined with the prototypical network for recognition, and the extracted patch images are compared with the prototypes at the corresponding positions at the same time. The sum of the metric values of the nine patch images is used to judge whether the object is detected. The time to detect each crop point is 1.22 seconds. The overall detection time is related to the number of objects in the input image. The detection time may also be related to different deep learning frameworks.

It should be noted that this study examined only the collected chili seedling crops. Considering human and material resources, it is not easy to collect data on other seeding crops. We tentatively believe that this solution can also be useful for UAV image detection in other seedling crops. Because UAV images are taken at a certain angle, there may be similarities between seedling crops of different types in an UAV image. It is possible to handle the two processes of segmentation and classification separately, following which, few-shot classifiers can be used for the segmented object patch image. The quality of segmentation depends on the development of the image segmentation field. The development results in the field of image segmentation can be absorbed in real time.

In this study, the 12 sliced categories can also be divided into two parts for two stages: meta-training and meta-testing. The categories of meta-test and meta-training stages may not overlap. Because it does not help complete the detection task, this setting is not used. If there are new categories to be identified in future studies, this dataset can be used as the source set, and the new categories can be used as the target set for the meta-testing stage.

Limitations

The proposed seedling crop detection framework completes the detection of chilies in UAV images with a small number of samples. However, this study has some limitations. There are two ways to recognize a new crop category after slicing the crop image into subcategories: first, the network model does not need to be retrained, and the new sliced categories are directly used in the meta-testing stage. The second method is to train all categories together. However, what impact this training will have on the classification and detection performance needs to be researched in future studies.

5. Conclusions

In this study, we propose a UAV image seeding crop detection framework, which is especially suitable for the detection of dense seedling crops from large-field optical UAV images. Considering that crops have similar characteristics on smaller scales, a subcategory slicing method is proposed, and a prototypical network is used to train the classification model. In the detection stage, a method for simultaneously extracting nine patches is used for object location. To evaluate the performance of the proposed method, it is tested on a UAV image dataset. The image dataset is generated using a UAV to collect images of seedling chilies. The method performs well in the image to be detected; the average Pacc and the average Uacc are 94.33% and 98.59%, respectively, with an error of 4.32%. The experimental results show that the proposed method can locate seedling chilies in UAV aerial images. This study provides a novel UAV image crop detection framework for the future, and can be applied to the detection and location of other crops. This detection framework can be improved in the future. For example, the detection search algorithm can be improved. In addition, advances in segmentation methods will also help improve the proposed method's performance.

Author Contributions: D.Z. and F.P. conceived the study. D.Z. designed the algorithm and wrote the original draft of this article. X.F. and W.L. supervised this research and revised this article. D.Z. analysed the results with help from F.P. and Q.D., J.W. assistant in data collecting. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the National Natural Science Foundation of China (No. 61973036), Yunnan Applied Basic Research Project of China (No. 201701CF00037), Guangdong Province Science and Technology Innovation Strategy Special Fund Project (No. skjtdzxrwd2018001) and Yunnan Provincial Science and Technology Department Key Research Program, China (Engineering) (No. 2018BA070).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Tao, C.; Mi, L.; Li, Y.; Qi, J.; Xiao, Y.; Zhang, J. Scene Context-Driven Vehicle Detection in High-Resolution Aerial Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7339–7351. [[CrossRef](#)]
2. Ji, H.; Gao, Z.; Mei, T.; Ramesh, B. Vehicle Detection in Remote Sensing Images Leveraging on Simultaneous Super-Resolution. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 676–680. [[CrossRef](#)]
3. Liu, Z.; Wang, H.Z.; Weng, L.B.; Yang, Y.P. Ship Rotated Bounding Box Space for Ship Extraction from High-Resolution Optical Satellite Images With Complex Backgrounds. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1074–1078. [[CrossRef](#)]
4. Zhang, Y.; Sheng, W.; Jiang, J.; Jing, N.; Wang, Q.; Mao, Z. Priority Branches for Ship Detection in Optical Remote Sensing Images. *Remote Sens.* **2020**, *12*, 1196. [[CrossRef](#)]
5. Zhou, M.; Zou, Z.; Shi, Z.; Zeng, W.-J.; Gui, J. Local Attention Networks for Occluded Airplane Detection in Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 381–385. [[CrossRef](#)]
6. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-scale Dataset for Object Detection in Aerial Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
7. Maes, W.H.; Steppe, K. Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture. *Trends Plant Sci.* **2019**, *24*, 152–164. [[CrossRef](#)]
8. Ochoa, K.S.; Guo, Z. A framework for the management of agricultural resources with automated aerial imagery detection. *Comput. Electron. Agric.* **2019**, *162*, 53–69. [[CrossRef](#)]
9. Hassler, S.C.; Baysal-Gurel, F. Unmanned Aircraft System (UAS) Technology and Applications in Agriculture. *Agronomy* **2019**, *9*, 618. [[CrossRef](#)]
10. Abdullahi, H.S.; Zubair, O.M. Advances of image processing in Precision Agriculture: Using deep learning convolution neural network for soil nutrient classification. *JMEST* **2017**, *4*, 2458–9403.
11. Bouachir, W.; Ihou, K.E.; Gueziri, H.-E.; Bouguila, N.; Belanger, N. Computer Vision System for Automatic Counting of Planting Microsites Using UAV Imagery. *IEEE Access* **2019**, *7*, 82491–82500. [[CrossRef](#)]
12. Mafanya, M.; Tsele, P.; Botai, J.; Manyama, P.; Swart, B.; Monate, T. Evaluating pixel and object based image classification techniques for mapping plant invasions from UAV derived aerial imagery: *Harrisia pomanensis* as a case study. *ISPRS J. Photogramm. Remote Sens.* **2017**, *129*, 1–11. [[CrossRef](#)]
13. Ma, L.; Li, M.; Ma, X.; Cheng, L.; Du, P.; Liu, Y. A review of supervised object-based land-cover image classification. *ISPRS J. Photogramm. Remote Sens.* **2017**, *130*, 277–293. [[CrossRef](#)]
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
15. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [[CrossRef](#)]
16. Xiong, X.; Duan, L.; Liu, L.; Tu, H.; Yang, P.; Wu, D.; Chen, G.; Xiong, L.; Yang, W.; Liu, Q. Panicle—SEG: A robust image segmentation method for rice panicles in the field based on deep learning and superpixel optimization. *Plant Methods* **2017**, *13*, 104. [[CrossRef](#)]
17. Chen, Z.; Wang, C.; Wen, C.; Teng, X.; Chen, Y.; Guan, H.; Luo, H.; Cao, L.; Li, J. Vehicle detection in high-resolution aerial images via sparse representation and superpixels. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 103–116. [[CrossRef](#)]
18. Malek, S.; Bazi, Y.; Alajlan, N.; AlHichri, H.; Melgani, F. Efficient Framework for Palm Tree Detection in UAV Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4692–4703. [[CrossRef](#)]
19. Ha, J.G.; Moon, H.; Kwak, J.T.; Hassan, S.I.; Dang, M.; Lee, O.N.; Park, H.Y. Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles. *J. Appl. Remote Sens.* **2017**, *11*, 042621. [[CrossRef](#)]

20. Wang, Y.; Zhu, X.; Wu, B. Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier. *Int. J. Remote Sens.* **2019**, *40*, 7356–7370. [[CrossRef](#)]
21. Fan, Z.; Lu, J.; Gong, M.; Xie, H.; Goodman, E.D. Automatic Tobacco Plant Detection in UAV Images via Deep Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 876–887. [[CrossRef](#)]
22. Hung, C.; Xu, Z.; Sukkarieh, S. Feature Learning Based Approach for Weed Classification Using High Resolution Aerial Images from a Digital Camera Mounted on a UAV. *Remote Sens.* **2014**, *6*, 12037–12054. [[CrossRef](#)]
23. Milioto, A.; Lottes, P.; Stachniss, C. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 41–48. [[CrossRef](#)]
24. Sa, I.; Chen, Z.; Popović, M.; Khanna, R.; Liebisch, F.; Nieto, J.; Siegwart, R. weedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming. *IEEE Robot. Autom. Lett.* **2018**, *3*, 588–595. [[CrossRef](#)]
25. Milioto, A.; Lottes, P.; Stachniss, C. Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2229–2235.
26. Bah, M.D.; Hafiane, A.; Canals, R. Deep Learning with Unsupervised Data Labeling for Weed Detection in Line Crops in UAV Images. *Remote Sens.* **2018**, *10*, 1690. [[CrossRef](#)]
27. Ocer, N.E.; Kaplan, G.; Erdem, F.; Kucuk Matci, D.; Avdan, U. Tree extraction from multi-scale UAV images using Mask R-CNN with FPN. *Remote Sens. Lett.* **2020**, *11*, 847–856. [[CrossRef](#)]
28. Poblete-Echeverría, C.; Olmedo, G.F.; Ingram, B.; Bardeen, M. Detection and Segmentation of Vine Canopy in Ultra-High Spatial Resolution RGB Imagery Obtained from Unmanned Aerial Vehicle (UAV): A Case Study in a Commercial Vineyard. *Remote Sens.* **2017**, *9*, 268. [[CrossRef](#)]
29. Donmez, C.; Villi, O.; Berberoglu, S.; Cilek, A. Computer vision-based citrus tree detection in a cultivated environment using UAV imagery. *Comput. Electron. Agric.* **2021**, *187*, 106273. [[CrossRef](#)]
30. Lin, Z.; Guo, W. Cotton Stand Counting from Unmanned Aerial System Imagery Using MobileNet and CenterNet Deep Learning Models. *Remote Sens.* **2021**, *13*, 2822. [[CrossRef](#)]
31. Tetila, E.C.; Machado, B.B.; Belete, N.A.; Guimarães, D.A.; Pistori, H. Identification of Soybean Foliar Diseases Using Unmanned Aerial Vehicle Images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2190–2194. [[CrossRef](#)]
32. Bullock, D.; Mangeni, A.; Wiesner-Hanks, T.; DeChant, C.; Stewart, E.L.; Kaczmar, N.; Kolkman, J.M.; Nelson, R.J.; Gore, M.A.; Lipson, H. Automated Weed Detection in Aerial Imagery with Context. *arXiv* **2019**, arXiv:1910.00652. Available online: <https://arxiv.org/abs/1910.00652> (accessed on 20 November 2019).
33. Dong, J.; Chen, Q.; Feng, J.; Jia, K.; Huang, Z.; Yan, S. Looking Inside Category: Subcategory-Aware Object Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1322–1334. [[CrossRef](#)]
34. Chen, T.; Lu, S.; Fan, J. S-CNN: Subcategory-Aware Convolutional Networks for Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2522–2528. [[CrossRef](#)] [[PubMed](#)]
35. Lake, B.M.; Salakhutdinov, R.; Gross, J.; Tenenbaum, J.B. One shot learning of simple visual concepts. In Proceedings of the 33rd Annual Conference of the Cognitive Science Society, Boston, MA, USA, 20–23 July 2011.
36. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. *Adv. Neural Inf. Processing Syst.* **2016**, *29*, 3630–3638.
37. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical networks for few-shot learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 4080–4090.
38. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
39. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv.* **2003**, *53*, 63. [[CrossRef](#)]
40. Ren, X.; Malik, J. Learning a Classification Model for Segmentation. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003. [[CrossRef](#)]