Wei and Wang *EURASIP Journal on Wireless Communications and Networking* (2019) 2019:150
https://doi.org/10.1186/s13638-019-1440-2

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Adaptive cascade single-shot detector on wireless sensor networks

Zhiyong Wei* and Fengling Wang

Abstract

The target detection model based on convolutional neural networks has recently achieved a series of exciting results in the target detection tasks of the PASCAL VOC and MS COCO data sets. However, limited by the data set for a particular scenario, some techniques or models applied to the actual environment are often not satisfactory. Based on cluster analysis and deep neural network, this paper proposed a new Statistic Experience-based Adaptive One-shot Network (SENet). The whole model solved the following practical problems. (1) By clustering the existing image classification dataset ImageNet, a common set of target detection datasets is formed, and a data set named ImageNet iLOC is formed to solve the object detection. The problem of single and insufficient quantities in the task. (2) We use cluster analysis on the size and shape of objects in each sample, which solves the problem of inaccurate manual selection of suggested areas during object detection. (3) In the multi-resolution training and prediction process, we reasonably allocate the size and shape of the suggested frame at different resolutions, greatly improve the utilization rate of the proposed frame, reduce the calculation amount of the model, and further improve the real-time performance of the model. The experimental results show that the model has a breakthrough in accuracy and speed (FPS reaches 54 in the case of a 3.4% increase in mAP).

Keywords: Cluster, Detection, Deep neural networks, Filter banks, Detection data sets

1 Introduction

Convolutional neural network (CNN) was widely used in the 1990s (such as model [1]), but with the rise of support vector machines in the field of computer vision, CNN entered a period of low tide. In 2012, the image classification model proposed by Krizhevsky et al. [2] demonstrated the revolutionary image classification accuracy in ILSVRC (ImageNet Large Scale Visual Recognition Challenge, ILSVRC) [3], rekindling people's interest in CNN. A series of image classification models based on CNN are continually proposed, and the image classification records of ILSVRC are also refreshed again and again. At the same time, the mean average precision (mAP) and detection speed of the target detection reference data set PASCAL VOC [4] and MS COCO [5] are also constantly increasing. Firstly, the success of selective search (SS) [6] and region proposal based volume and neural networks (R-CNN) [7] has driven advances in object detection. Although R-CNN was very time consuming to propose, the cost of the detection model was greatly reduced by sharing the convolution between the proposed regions [2, 7]. Typical research results, such as Fast R-CNN, have achieved real-time target detection speeds without the time-consuming recommendations of regional recommendations, using target detection models with extremely deep convolutional neural networks [2]. However, the time-consuming problem of regional recommendations remains the performance bottleneck of the most advanced target detection systems. Next, region proposal networks (RPN) using CNN instead of selective lookups is proposed [7]. RPN shares a partial convolutional layer with the most advanced target detection network, and by testing the shared convolution, the cost of the calculation suggestion box is further reduced. However, the entire detection model (Faster R-CNN) needs to train a proposed network and a detection network [7], which is still too cumbersome, inefficient, and not easy to optimize relative to the detection model of a single network.

YOLO regards target detection as a single regression problem. It first extracts the input image through a

* Correspondence: ncvtweizhiyong@163.com
Nanning College for Vocational Technology, Nanning 530008, Guangxi, China

traditional convolutional neural network to form an S × S grid (e.g., 7 × 7). Each grid produces two bounding boxes of different sizes and shapes (7 × 7 × 2 = 98) relative to the original image, each bounding box representing the coordinates of a potential object and the probability of belonging to a certain category.

The SSD model is similar to YOLO. The main difference is that the SSD architecture combines multiple feature maps of different resolutions in the neural network for target prediction, naturally processing objects of different sizes, and improving detection quality [8]. However, the single dataset, poor integration, low speed and accuracy, and difficulty in optimization are still many of the problems faced by the object detection model. This chapter proposes an end-to-end, single neural network target detection model, which mainly has the following contributions:

By applying the clustering analysis algorithm to the existing ImageNet for image classification, a set of general methods for making image classification data sets into target detection data sets is formed, which solves the problem of single and insufficient samples in object detection tasks.

The paper uses cluster analysis to determine the size and shape of the bounding box in each sample, and obtains the prior knowledge of the size and shape of the bounding box of the object in the data set. Based on these priors, the design of the model's border frame is guided, and the object detection process is solved manually. Design suggestions for areas that are not accurate.

In the multi-resolution training and prediction process, this paper greatly improves the utilization rate of the proposed frame by reasonably assigning the size and shape of the suggestion frame of different resolution detection layers, and at the same time, reduces the calculation amount of the model and further improves the calculation. The real-time and accuracy of the model.

1.1 Related works

In this part, we first introduce the neural network-based target detection and recognition methods. On this basis, the plant anomaly detection technology was reviewed, and the research progress of false positive detection technology was reviewed.

1.2 Image-based object detection and feature extractors

In recent years, visual media gained through the Internet has proliferated. A large amount of data brings new opportunities and challenges to the application of neural networks. Since Alex Net [2] first applied the convolutional neural network (CNN) to image classification tasks in the ImageNet Large Scale Visual Identity Competition (ILSVRC-2012) [3], it consists of eight layers. CNN demonstrated superior performance compared to traditional manual computer vision algorithms. Therefore, in recent years, several deep neural network structures have been proposed to improve the accuracy of the same task.

Object detection and recognition is an important issue in recent years. In the case of detecting specific categories, earlier applications focused on image classification from object-centric [9]. The goal is to categorize images that may contain objects. However, the new main paradigm is not only to classify and accurately locate objects in an image [10]. Therefore, current prior art object methods for object detection are primarily based on deep CNN [3]. They are divided into two phases and a one-stage approach. Two-stage methods are often associated with region-based convolutional neural networks, such as faster R-CNN [7], region-based fully connected networks (R-FCN) [11]. In these frameworks, the region proposal network (RPN) generates a set of candidate object locations in the first level, and the second level uses CNN to classify each candidate location into one of the classes or backgrounds. It uses a deep network to generate features for backward use by the RPN to extract recommendations. In addition to the system based on regional recommendations, a first-level framework for object detection has been proposed. Recent SSDs [8], Yolo [12] and Yolo v2 [13] have shown promising results, resulting in a real-time detector similar to the accuracy of a two-stage detector. In the past few years, it has also been demonstrated that deeper neural networks achieve higher performance than simple models in image classification tasks [3]. However, with significant performance improvements, the complexity of deep structures has also increased, such as VGG [14], ResNet [15], GoogleNet [16], ResNext [17], DenseNet [18], dual path network [19], and Senet [20]. As a result, deep artificial neural networks often have much more trainable model parameters than the number of samples they accept [21]. Although a large number of data sets are used, neural networks tend to over-fitting [1]. On the other hand, several strategies have been applied to improve performance in deep neural networks. For example, increasing the number of samples increases the data [22], weighting regularization to reduce model overfitting [23], random discarding off-activation [24], and batch normalization [25]. While these strategies have proven to be effective in large networks, the lack of data or category imbalances remains a challenge for several applications. There is no specific way to understand the complexity of artificial neural networks for

their application to any problem. Therefore, the importance of developing a strategy specifically designed for applications that include limited data and class imbalance issues. Moreover, depending on the complexity of the application, today's challenge is to design a deep learning approach that can perform complex tasks while maintaining lower computational costs.

1.3 Anomaly detection in plants

The problem of plant diseases is an important issue directly related to people's food safety and welfare. Diseases and pests affect food crops, which in turn cause significant losses in the peasant economy. The effects of disease on plants are becoming challenging in crop protection and healthy food production. Traditional methods for identifying and diagnosing plant diseases depend primarily on the visual analysis of experts within the area, or in the laboratory. These studies often require high expertise in the field, in addition to the probability of not successfully diagnosing a particular disease, thus leading to erroneous conclusions and treatment (proposed 2018). In these cases, in order to obtain quick and accurate decisions, automated systems will provide efficient support to identify diseases and pests of infected plants [26, 27]. Recent advances in computing technology, particularly graphics processing units (GPUs), have led to the development of new image-based technologies, such as efficient deep neural networks. The application of deep learning has also expanded into the field of precision agriculture, and it has also shown satisfactory results while solving complex problems. Some applications include disease identification for several crops such as Cole [26], apples [28], bananas [29], wheat [30], and cucumbers [31]. The CNN-based approach constitutes a powerful tool that is used as a feature extractor in multiple jobs. Mohanty et al. [27] compared two CNN architectures, Alexnet and Googlenet, to identify 14 crop species and 26 diseases using large disease databases and healthy plants. Their results show a system that effectively classifies images containing specific diseases into crops that use transfer learning. However, the disadvantage of this work is that its analysis is based only on images collected in the lab, not in actual field scenarios. Therefore, it does not cover all the changes contained there. Similarly, Sladjevic et al. [32] identified 13 healthy leaf plant diseases using the Alexnet CNN architecture. They used several strategies to avoid overfitting and improve classification accuracy, such as data enhancement techniques, to increase the size of the data set and improve efficiency while training CNN. The average accuracy of the system is 96.3%. Recently, Liu et al. [28] proposed a method for apple leaf disease identification based on a combination of Alexnet and GoogleNet architectures. The system was trained to identify four types of apple leaf disease using an image data set collected in the laboratory with a total accuracy of 97.62%. In [33], ferencinos evaluated various CNN models to detect and diagnose plant diseases using leaf images of healthy and infected plants. The system is capable of classifying 58 different plant/disease combinations from 25 different plants. In addition, the experimental results show interesting comparisons when using images collected in the laboratory compared to images collected in the field. The use of two types of images gives promising results with an optimum accuracy of 99.53% given by the VGG network. However, when images acquired in the field are used for testing rather than laboratory images, the success rate is significantly reduced. In fact, according to the author, this proves that image classification under real field conditions is more difficult and complicated than using images collected in the laboratory.

Although the above work has achieved good results in the identification of plant diseases, the challenges of complex field conditions, infection changes, various pathologies in the same image, and surrounding objects have not been studied. They mainly use images acquired in the lab, so they cannot handle all the situations that occur in real scenes. Moreover, they are all based on the method of disease classification. In contrast, Fuentes et al. [26] proposes a system that can successfully detect and locate nine Cole pests and diseases using images collected in the field, including actual cultivation conditions. This approach differs from other methods in that it generates a set of bounding boxes that contain the location, size, and category of the disease and/or pest in the image. This work examines different meta-architectures and CNN feature extractors to identify and locate suspicious regions in an image. The results show that the performance of the method reaches 83%. However, the system has some difficulties that make it impossible to achieve higher performance. They mentioned that due to the lack of samples, some highly variant classes are often confused with other classes, resulting in false positives or low precision. According to the idea in [26], our current work aims to solve the above problems and improve the results by focusing on false positives and class imbalances. On the other hand, our method has studied several techniques to make the system more robust to inter- and intra-species changes in Cole pests and diseases.

2 Our methods

2.1 Classified dataset to test dataset

The target detection data set is costly to produce (you need to label the number of objects in the image, the size, shape, location, and type of each object), and the

public can obtain less (more famously, only PASCAL VOC and COCO data sets). The problem of single type of data set severely limits the accuracy and application range of the target detection model. On the other hand, because the image classification data set is relatively easy to make (only the type of the object needs to be labeled), there are many kinds of types. The publicly available data set is used to train the image classification model. More famous are MNIST [34], CIFAR [35], ImageNet [3], Youtube-8 M [36], Open Image [37], etc. These data sets contain nearly 10,000 kinds and hundreds of millions. At the same time, some well-known image classification models achieved very high accuracy (even 100%) on the relevant data sets. Do we have a way to automatically add location tags to these well-known classification datasets? In the field of machine learning, there are many unsupervised algorithms that are very suitable for finding the shape of randomly distributed data. Spectral clustering is one of them [38]. Spectral theory is the study of how the properties of a graph are described by several easily calculated quantities. Spectral clustering of a graph is an important tool for describing graphs. The usual method is to encode the graph into a matrix and then calculate the eigenvalues of the matrix (also called spectrum spectrum) [38]. In other words, spectral clustering uses a weighted adjacency matrix and its spectral map to analyze the data by graph segmentation [39]. Therefore, spectral clustering can achieve more powerful data representation in the feature space by using the main components in the data revealed by the spectrum, thus facilitating data clustering.

The object detection data set usually consists of the center coordinates of the bounding box of the object, the length and width, and the category label of the object, which are usually represented by $\{cx, cy, w, h, category\}$. This chapter uses spectral clustering algorithm and ResNet [17] image classification model to propose a method to make the classification data set into object detection data set. The algorithm first uses spectral clustering to obtain the contours (shapes) of the objects in the image, and calculates the position coordinates $\{cx, cy, w, h\}$ of each object; using ResNet to classify the objects in the image to obtain a confidence that it belongs to a certain category. Degree (confidence). The algorithm flow is divided into the following stages to complete the production of the test data set:

1) Obtain image left and lower edge feature coordinate vectors $= \{\vec{l}_1, \vec{l}_2, \ldots, \vec{l}_m\}; D = \{\vec{d}_1, \vec{d}_2, \ldots, \vec{d}_n\}$, respectively, of size $m \times n$.
2) Cluster analysis of the pixels of the image using spectral clustering to obtain a rough outline of each object $O = \{O_1, O_2 \ldots O_M\}$ in the image, and a pixel coordinate vector O_j of each object (cluster) $C^j = \{\vec{c}^j_1, \vec{c}^j_2 \ldots \vec{c}^j_K\}$.
3) Calculate the object coordinate vector set C^j based on the formula (1–4), the four coordinates of the nearest and farthest distance of each coordinate vector to the edge set $\{L, D\}$ $T_j = \{\vec{t}^j_u, \vec{t}^j_d, \vec{t}^j_l, \vec{t}^j_r\}$, using coordinates (x_1, y_1), (x_2, Δ_y), (Δ_x, y_3), (x_4, y_4) indicates.

$$\vec{t}^j_u = \max_{k \in K} \text{distance}\left(D, \vec{c}^j_k\right) \tag{1}$$

$$\vec{t}^j_d = \min_{k \in K} \text{distance}\left(D, \vec{c}^j_k\right) \tag{2}$$

$$\vec{t}^j_l = \min_{k \in K} \text{distance}\left(L, \vec{c}^j_k\right) \tag{3}$$

$$\vec{t}^j_r = \max_{k \in K} \text{distance}\left(L, \vec{c}^j_k\right) \tag{4}$$

4) After getting the four vertices of the object, we calculate the width, height, and center coordinates of the bounding box based on the formula (5–7):

$$w = x_4 - \Delta_x \tag{5}$$

$$h = y_1 - \Delta_y \tag{6}$$

$$\left(c_x, c_y\right) = \left(\Delta_x + w, \Delta_y + h\right) \tag{7}$$

5) Finally, we use ResNet to calculate the category confidence of the object in the rectangle. If the confidence of this object belongs to a category is greater than 0.85, then the correct label of the algorithm output sample (ground-truth label) $\{(c^j_x, c^j_y), w, h, c\}$. Otherwise, ignore this object. See Fig. 1 for the specific process and a schematic diagram of the algorithm.

Algorithm 1 Generating object detection dataset based on spectral clustering

Input: Classification sample X and the size of the dataset is n.
Output: Detection object D
Function CLASSIFICATIONTODETECTION(X)
Initialization $k_{cluster} = 6$
for $i = 1$ to n **do**
Compute the left and bottom coordinates of sample X_i
Update the coordinate set $\{L, D\}$
Spectral clustering for detection object O based on sample X_i
for $j = 1$ to O_M **do**
Compute the upper vertex t_u^j, bottom vertex t_d^j, left vertex t_l^j and right vertex t_r^j of object O_j
Update the object coordinate $T_j = \{\vec{t}_u^j, \vec{t}_d^j, \vec{t}_l^j, \vec{t}_r^j\}$
Compute the width w and height h of the anchor box
Compute the center coordinate (c_x^j, c_y^j) of the anchor box
Update the positioning information of O_j as $b_j = \{(c_x^j, c_y^j), w, h\}$
Compute the parameter confidence of object O_j as $c = ResNet(b_j)$
if $\max_c(c_i) > 0.85$ **then**
Compute the ground truth box of object O_j as $gtb_j = \{(c_x^j, c_y^j), w, h, c\}$
Update the detection dataset with gtb_j
else
continue
end if
end for
end for
return the new detection dataset D
end function

2.2 Boundary box clustering (k-means++)

At present, the recommended boxes of classical object detection models are based on manual experience, and the size and shape of the boundary boxes are set manually. In reality, the manual designed boundary boxes are usually neither flexible nor robust [3, 7, 9]. For this reason, this paper uses k-means++ algorithm to cluster the shape of the object in the sample, so as to objectively obtain the shape distribution of the object in the sample. Standard k-means++ calculates the distance from each element to the cluster center based on the Euclidean distance. Because the distance between each element and the cluster center varies greatly among objects of different sizes, using absolute distance to calculate the shape of the object will make k-means++ unable to converge correctly. Therefore, this paper uses formula (8) instead of the standard Euclidean distance calculation method to complete the clustering analysis of object shape, in which box is the sample to be clustered (including the width and height of the object), centroid is the center of the current clustering, and the function of intersection over union (IoU) outputs the overlap ratio between the object and the clustering cluster. Figure 2a shows that when k-means++ chooses different K values, $k = 6$ is selected to balance the speed and the overlap rate of IoU. Figure 2b shows the distribution of objects of different sizes and shapes in the original image by

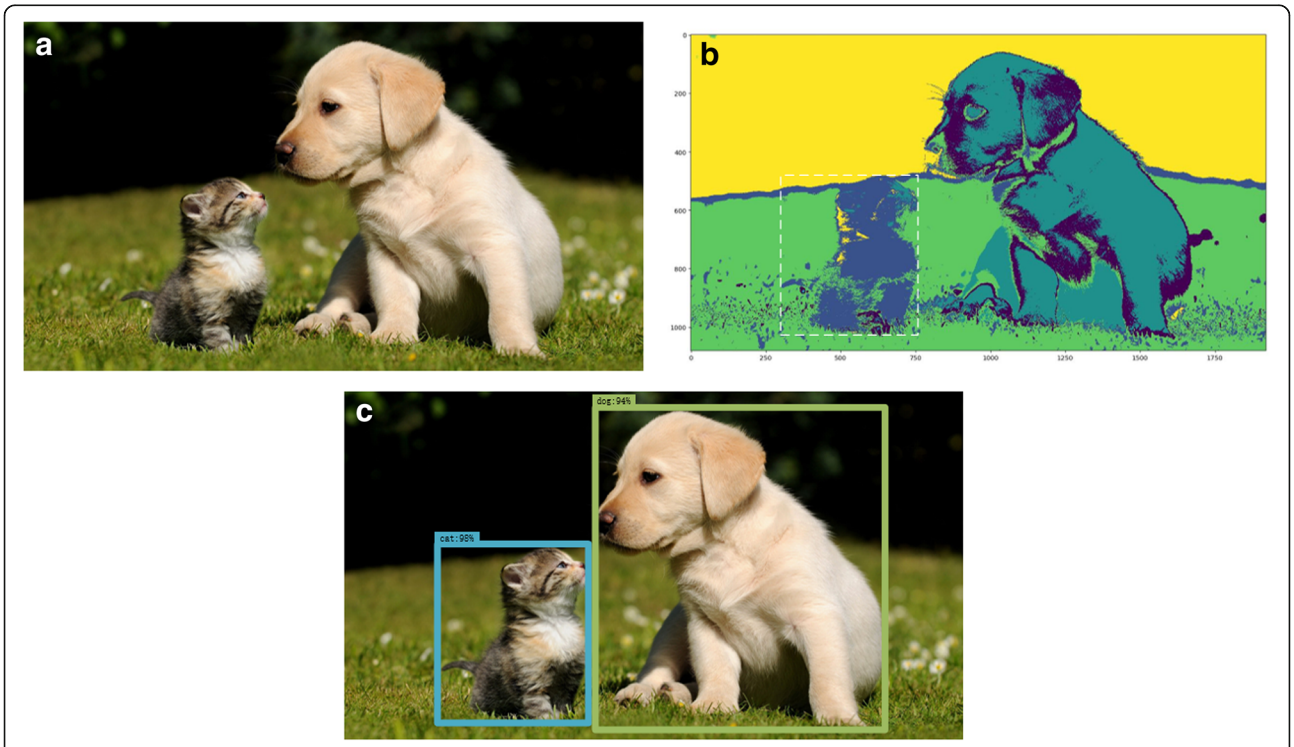

Fig. 1 A monitoring system with wireless sensor networks. Figure 1 shows a monitoring system with wireless sensor networks. **a** Original image. **b** Anchor box defining. **c** Object recognition

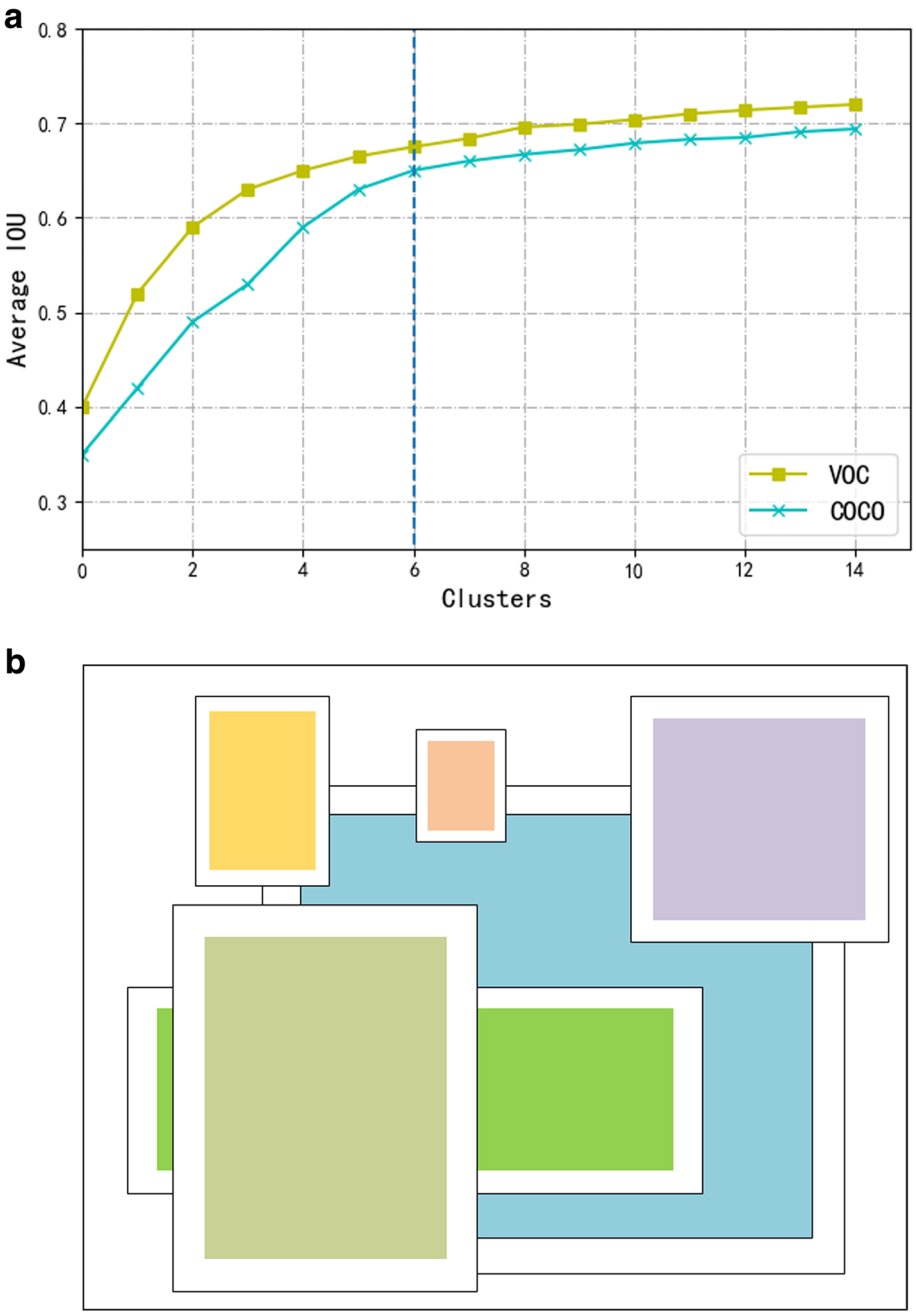

Fig. 2 A general overview of the approach we propose. Figure 2 shows the entire proposed system. Input images of any size are trained in our primary diagnostic unit, which generates boundary boxes and their location and category of infected areas in the image. The secondary diagnostic unit uses the bounding box as an input, and the secondary diagnostic unit independently trains the CNN filter bank to reduce the number of false positives generated by the primary unit. Both systems are further integrated into the level and location. K-means++ clusters the border shapes of training samples from PASCAL VOC and COCO datasets. **a** It shows that when k-means++ chooses different *K* values, $k = 6$ is chosen to balance the speed and the overlap rate of the IOU. **b** K-means + + clustering results show that thin and high boundaries account for the majority of the samples

clustering the ImageNet and COCO data sets. We find that the shape of objects is different from the default box set manually. The boundary boxes of high and thin shapes occupy the majority of shapes. Again, it proves the necessity of clustering the shape of samples to obtain the priori shape of objects.

$$d(\text{box}, \text{centroid}) = 1 - IoU(\text{box}, \text{centroid}) \quad (8)$$

Figure 3a shows the method of setting default boxes manually. Due to lack of objective analysis, it can not effectively improve the phenomenon of overlapping areas

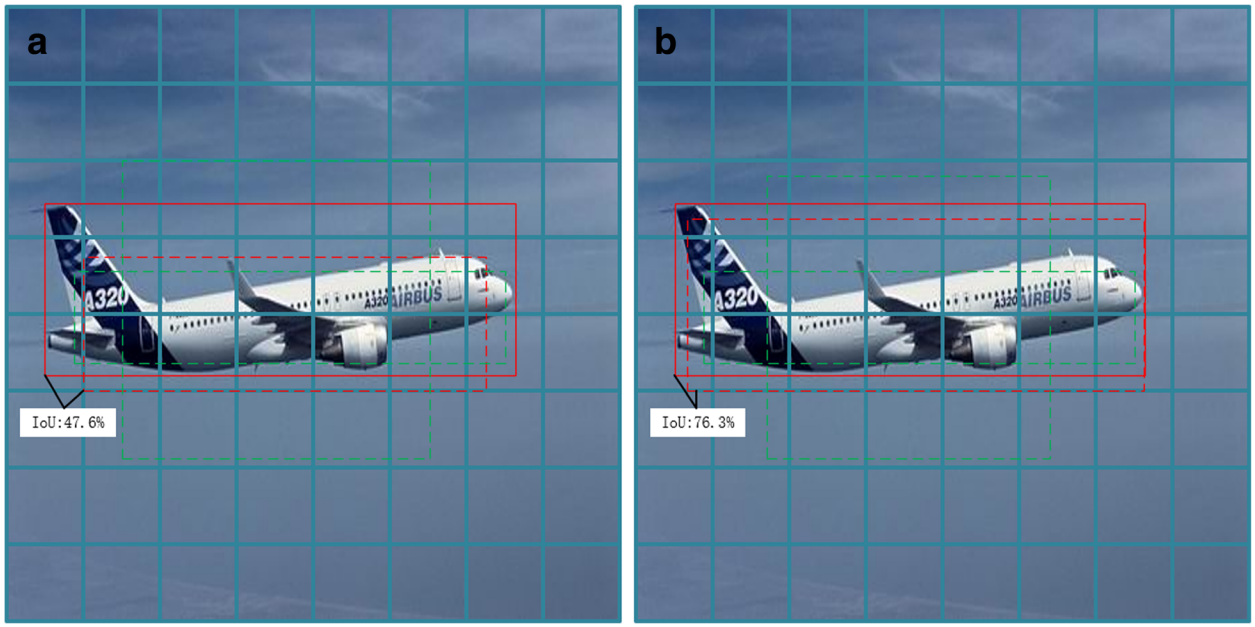

Fig. 3 Primary diagnosis unit for bounding box detection. Figure 3 shows the process by which the primary diagnostic unit detects suspicious areas containing disease and pests in the input image. It is similar to Fuentes et al. (2017b). Compared the manual setting of boundary boxes with the k-means++ clustering generation of boundary boxes, in which the red solid border is the correct border and the red dotted border is the highest border in IoU of different methods of generating boundary boxes. **a** The size and shape of the default boundary box set manually are usually not able to obtain high IoU, and **b** the prior knowledge of the boundary box is obtained from tens of millions of samples using k-means++ (thin, high borders account for the majority)

of interest. The red solid border is the correct border, and the red dotted border is the highest border of IoU for different methods of generating boundary boxes. The average IoU of default boxes manually designed is only 47.6%. Figure 3b shows that after getting the priori boundaries of object shape and size from a large number of samples by clustering analysis, we can reduce the square boundaries, increase the boundaries of thin and high edges appropriately, effectively improve the proportion of IoU, and further improve the convergence speed and detection accuracy of the model.

It is different from K-means clustering algorithm used in Yolo and DSSD. K-means++ has better stability and efficiency in clustering results. Firstly, in order to ensure the high detection speed of the model, the default box size and shape in the model are usually taken as parameters, which are set up when the model is initialized, and no changes are made. This requires that the clustering results on the training set must be as stable as possible. However, K-means clustering algorithm is very sensitive to the initialization of parameters of the algorithm. Each randomly selected K clustering centers will produce completely different clustering results, which confuses researchers in choosing the priori shape of the boundary box. In addition, if the selection of clustering centers is inappropriate, the clustering results will be quite different, and even some results can not reflect the characteristics of the data set. Cluster centers of k-means++ are divided in turn from least to most. Each time the elements farthest from the current cluster centers are selected as new cluster centers, which reduces the uncertainty of random selection of k-means, improves the speed and accuracy of clustering, and provides a better basis for the selection of priori boxes for the detection model.

2.3 Model training

During training, the data input to the classifier includes the feature mapping of the nth detection layer, the category of the object in the correct border, and the probability that the object belongs to a certain category. The data input to the detector includes the feature mapping of the nth detection layer, the position coordinates of the correct and priori borders, and the offset errors of the output priori and the correct borders. The reality is that there are many differences between the IoU of each priori box and the correct border. We use two methods to judge a priori box as a positive sample: (1) the highest priori box overlapping the correct border box IoU, or (2) the IoU of a priori box and any correct border is greater than 0.7 (a correct border can be used as a label for multiple prioris). Although the second condition is sufficient to determine the positive sample, we still use the first condition, because in some rare cases, the second condition may not find the positive sample (for example, an

object is too small or too large). If the IoU ratio of a non-positive sample to all correct borders is less than 0.3, then we consider some prior boxes as negative samples. Finally, we discard the priori boxes that do not contribute to training, neither positive samples nor negative samples. The training method of the model originates from the multi-task minimization objective function, but extends to the category that can recognize multiple objects. The whole objective loss function is the weighting of detection loss (reg) and classification loss (cls):

$$L(x,c,p,g) = \frac{1}{N}\left(L_{cls}(x,c) + \alpha L_{reg}(x,p,g)\right) \tag{9}$$

Where x is the feature mapping of the output of different detection layers, c is the category of objects in a priori box d, p is the coordinate of the prediction box, and g is the correct border coordinate value. N is the number of matching priori frames. If $N = 0$, the loss is set to 0, and alpha controls the weight of detection error. Similar to Faster R-CNN, our regression method calculates the center $(\hat{t}_i^x, \hat{t}_i^y)$, width $(\hat{t}_i^w)$, and height $(\hat{t}_i^h)$ of the prediction *box* (p) and the center (t_j^x, t_j^y), width (t_j^w), and height (t_j^h) of the prediction box (p), respectively. Our regression method calculates the offsets between the center; the detection loss is based on the coordinate vector of the prediction box (p) calculated by smooth L1 based on the offset between $\{\hat{t}_i^m\}$ and the coordinate vector t_j^m of ground truth box g; AMME can train the model from end to end.

$$L_{reg}(x,p,g) = \sum_{i\in Pos}^{N}\sum_{m\in\{c_x,c_y,w,h\}} x_{ij}^k \mathrm{smooth}_{\mathrm{L1}}\left(\hat{t}_i^m - t_j^m\right) \tag{10}$$

Where:

$$\hat{t}_i^{c_x} = \frac{p_i^{c_x} - d_i^{c_x}}{d_i^w}, \hat{t}_i^{c_y} = \frac{p_i^{c_y} - d_i^{c_y}}{d_i^h}$$

$$\hat{t}_i^w = \log\frac{p_i^w}{d_i^w}, \hat{t}_i^h = \log\frac{p_i^h}{d_i^h}$$

$$t_j^{c_x} = \frac{g_j^{c_x} - d_i^{c_x}}{d_i^w}, t_j^{c_y} = \frac{g_j^{c_y} - d_i^{c_y}}{d_i^h}$$

$$t_j^w = \log\frac{g_j^w}{d_i^w}, t_j^h = \log\frac{g_j^h}{d_i^h}$$

The classified loss function is as follows:

$$L_{cls}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^p \log\left(\hat{c}_i^p\right) - \sum_{i\in Neg}\log\left(\hat{c}_i^0\right) \tag{11}$$

Where:

$$\hat{c}_i^p = \frac{\exp\left(c_i^p\right)}{\sum_p \exp\left(c_i^p\right)}$$

The model can use the error function above to optimize the proposed regions generated by all the prior boxes, but this will be biased toward negative samples, because their number of samples dominates. Therefore, in the training process, we use the min-batch method to randomly select 128 recommended areas at a time, and force the proportion of positive and negative samples to be kept at 1:1. If there are less than 64 positive samples in an image, the negative samples are filled in small batches. We randomly initialize all the new multi-resolution detection layers so that their parameters obey the Gauss distribution with zero mean and 0.1 variance. The basic network layer is initialized by pre-training the ImageNet classification model VGG-16.

3 Results

The hardware of the experiment is accomplished on a Dell server and equipped with two GTX-1080Ti GPUs. The operating system is Ubuntu 16.464 bits, which runs the Tensor-flow deep learning framework, and uses Tensor-board to monitor the training process. All the experimental results are based on VGG16 and trained in advance on ILSVRC datasets. Target detection training set and test set are Passcal VOC 2007, 2012, COCO, and our ImageNet iLOC data set based on Section 3.1. We use the AMME optimizer proposed in chapter 4 to fine-tune the model. The default parameters are learning_rate = 0.001, beta 1 = 0.9, beta 2 = 0.999, and = 1e-08. The learning rate of different data sets is slightly different from the setting of parameters of beta 1 and beta 2, which will be described in detail later.

3.1 Experimental results in PASCAL VOC 2007

On this data set, our SENET method is compared with SSD, YOLO, and Faster R-CNN. The data set used in this section includes PASCAL VOC 2007, training set in 2012, and verification set in PASCAL VOC 2007 and 2012, totaling 16,551 images. The test set uses PASCAL VOC 2007 test, including 4952 images. In the first 40 K iterations of the model, AMME uses learning_rate = 0.001, beta 1 = 0.9, beta 2 = 0.999, Euro = 1e-08, then reduces learning_rate = 0.0007, beta 1 = 0.75, beta 2 = 0.777, Euro = 1e-08, and then iterates 20 K. Table 1 shows that SENET's accuracy when using 300*300 as input has exceeded that of SSD model with the same size as input. This again shows that in Section 5.2.1, we

Table 1 The results of PASCAL VOC 2007 test set

Method	mAP	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV
Faster R-CNN	73.2	76.2	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
YOLO V2	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7
SSD300	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD512	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
EAO300	74.9	75.7	80.1	74.3	66.6	53.6	82.0	83.6	85.7	58.6	78.2	75.9	83.7	83.3	82.7	77.2	49.9	73.9	75.3	82.6	74.6
EAO512	78.1	85.7	85.4	78.8	71.3	55.4	84.9	87.3	86.9	59.2	82.8	74.3	85.9	87.1	85.7	81.9	54.5	78.7	74.1	84.9	76.3

can better match the correct border by clustering the boundaries of the training set, so as to improve the accuracy of the model. When the image training model of 512 × 512 is input more, SENET's mAP easily surpasses Faster R-CNN (mAP reaches 78.1%, 5.9%, 1.3% higher than SSD512). Moreover, most of its high confidence tests are correct, and the recall rate is about 85–90%. Compared with the R-CNN step training using two stitching methods, our SENET model directly regresses the shape of the object and the category of the classified object, so it is easier to train and optimize, so it has less detection error.

Table 1 shows the results of PASCAL VOC2007 test set. Among them, the input image sizes of model SSD, YOLO, and Faster R-CNN are 512 × 512. Our SENET model uses two sizes (300 × 300) and (512 × 512) to compare with each baseline model.

To illustrate the performance of two SENET models with different input sizes in more detail, we use tensorboard detection and analysis tools from Google tensorflow to analyze the training process of the model. As can be seen from Fig. 4 because of our multi-resolution detection layer, using more efficient priori box generation method can get better IOU value, which is conducive to accelerating the convergence speed and improving the accuracy of the model. Compared with SSD (right side of Fig. 4), the convergence speed of the model is faster (about 60 K iteration model has converged), and mAP is higher than SSD. At the same time, we can also see that SENET's convergence is more stable than SSD's.

Most detection models detect smaller objects with worse performance than larger objects, mainly because after multi-layer convolution, the feature mapping of the smallest objects at the top level may not have any information left. In Fig. 5 based on the clustering results, we use different shape and number of priori boxes in different detection layers, which makes the model less sensitive to the size of boundary boxes than SSD. The experimental results also show that our SENET model has better performance and stronger robustness than SSD when detecting smaller objects.

3.2 Experimental results of MS COCO

MS COCO 2015 has 91 classifications, each of which has about 10,000 samples, and each sample (image) has about one to six objects. To further validate the proposed method, we train SENET 300*300 and 512*512 models on MS COCO datasets. Because COCO data sets have many kinds of objects, many objects to be detected in a single sample and small objects to be detected, the gradient oscillation of the model is large when it starts training. First, the parameters of learning_rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$, batch-size = 128 are trained

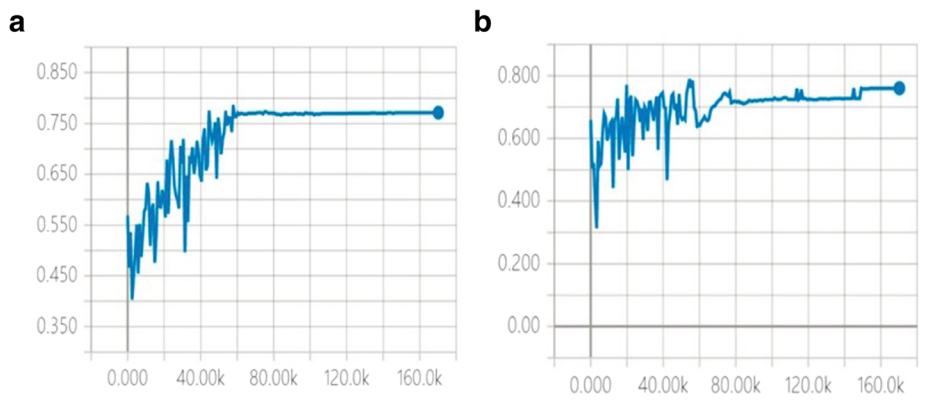

Fig. 4 VOC2007 test set compares the convergence speed and the stability of training convergence of SENET512 (4**a**) and SSD512 (4**b**) models

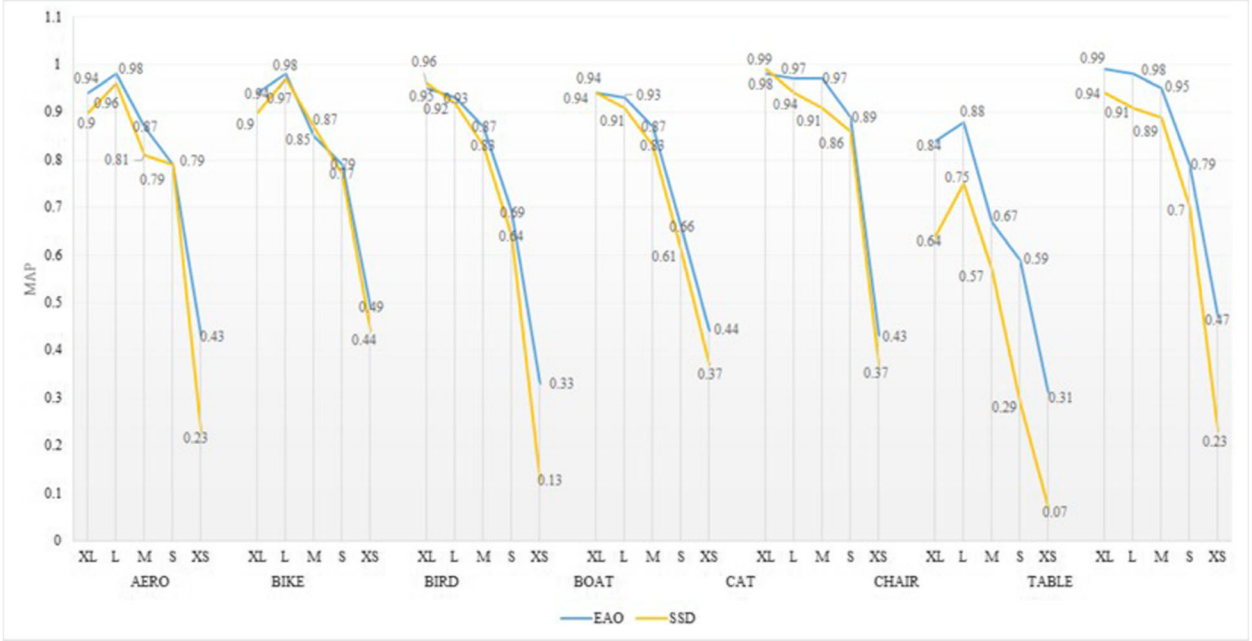

Fig. 5 The results of the bounding box detector demonstrate an imbalance between the classes. Each column represents a comparison of the number of borders for each class using different intersections within a 10% to 90% joint threshold range. SENET and SSD models are used to compare the accuracy of identifying objects of different sizes. The overall performance of SENET model is better than that of contrast model, and the recognition accuracy of chair with sparse structure is much higher than that of SSD

and iterated 140 K times. Then, the parameters of ADAM are adjusted to learning_rate = 0.0009, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$, batch-size = 64 continue to iterate 50 K times.

Following the strategy mentioned in Section 5.3.2, we cluster the size and shape of objects in COCO datasets and use smaller priori boxes for all detection layers. Then, based on the method of Section 5.3.3, the shape and number of cell priori frames in different detection layers are designed. We evaluated the mAP value with IOU < [0.5:0.05:0.95] (standard COCO measurement method, simply quoted as mAP@[.5,.95]) and mAP@0.5 (PASCAL VOC measurement method).

Table 2 shows the test results of each model on test-dev2015. Similar to what we observed on PASCAL VOC datasets, SENET 300*300 outperforms Faster R-CNN and YOLO in mAP@0.5 and mAP@0.95, and is very close to SSD512. However, whether SENET300 or SENET3512, its mAP@0.5 is significantly better than SSD and YOLO. We speculate that this is because the size of objects in MS COCO datasets is too small, and SSD and YOLO models are not good at locating many small objects accurately, which leads to model failure. The experimental results also show that by increasing the size of the input image to 512*512, SENET is more accurate than all baseline models in both test criteria. The experimental results also show that SENET512 model is better than ION [162]. It is a multi-size version of Fast R-CNN, which uses cyclic neural network to explicitly simulate feature context. In Fig. 6 some detection results of MS COCO test-dev using 512*512 model and convergence process of SENET model boundary box are shown.

4 Discussion

Firstly, aiming at the problem of single sample and high production cost of current object detection data set, based on clustering analysis algorithm and image classification model, this paper proposes a method of integrating classification data into detection data set, and makes ImageNet iLOC detection data set from ImageNet classification data set. The experimental results show that the accuracy of the model can be improved by 4.3% by using the proposed ImageNet iLOC detection data set and

Table 2 Testing results of MS COCO show that SENET model performs well in small objects

Mode	Boxes	mAP@[0.5:0.95]	mAP@0.5	mAP@0.75
Faster R-CNN	RPN 300	21.9	42.5	21.9
ION	–	22.4	42.7	18.7
YOLO	97	21.4	43.8	19.6
SSD512	8735	25.5	44.9	22.7
SENET300	5965	21.7	45.1	24.6
SENET512	5965	*26.9*	*47.6*	*28.2*

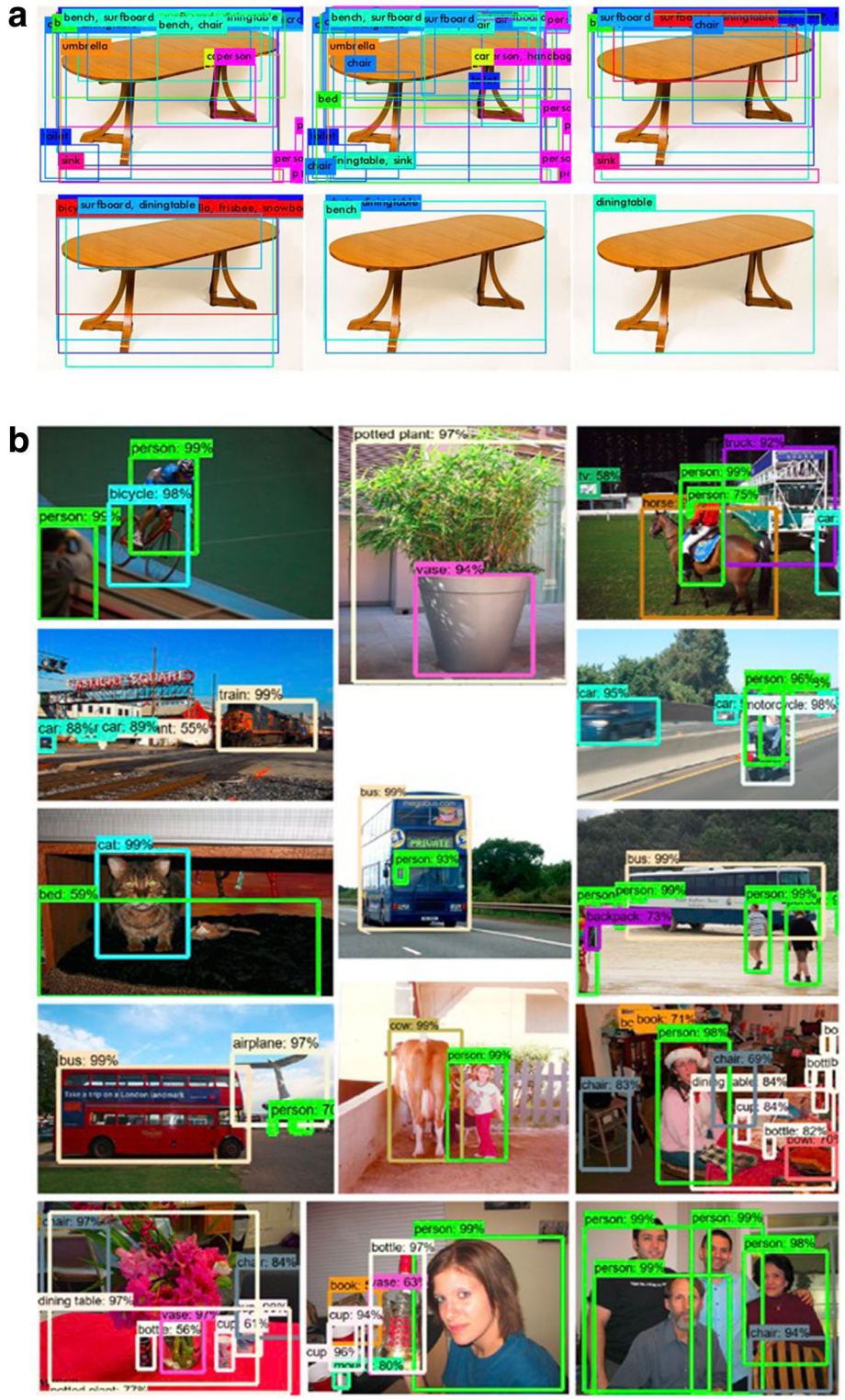

Fig. 6 Some false positives produced in the primary diagnostic unit. **a** The ulcer sample was detected as plague. **b** The gray mold sample was detected as ulcer disease. **c** The low temperature sample was detected as ulcer disease. The detection effect of MS COCO dataset is shown, **a** the convergence process of boundary box location and object classification, and **b** the detection effect of SENET512 on COCO dataset

continuing to train the model. Then, aiming at the problem that the current popular object detection model can not design the default box manually and accurately, this chapter uses K-means++ clustering algorithm to cluster the shape of the object in the sample, and obtains the shape distribution of the object. Based on this priori knowledge, the designed priori box model and ground true box IOU are higher, which greatly improves the convergence speed of model training. Finally, according to the characteristics of different multi-resolution detection layers corresponding to different size regions of the original image, we carefully design the size and shape of the priori box of each detection layer to form an empirical adaptive single network detection model SENET. In the experimental part, I continue to train SENET model with ImageNet iLOC data set. The results show that the accuracy of the SENET model proposed in this chapter is improved by about 3–4% on average compared with other benchmark models. Future work will further improve the practicability of the model. For example, the model will start from the video of a specific scene (e.g., unmanned driving), real-time object detection in video content, and ensure high speed and accuracy.

Abbreviations

CNN: Convolutional neural network; fps: Frame per second; GPU: Graphics processing units; ILSVRC: ImageNet Large Scale Visual Recognition Challenge; IoU: Intersection over union; mAP: Mean average precision; R-CNN: Region proposal based CNN; RPN: Region proposal networks

Acknowledgements

Subsidies from the Training Program for 1000 Young and Middle-aged Key Teachers in Guangxi Colleges and Universities, Nanning Special Expert Project Funding.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Authors' contributions

AF designed the study, performed the experiments, and data analysis, and wrote the paper. DP and SY advised on the design of the system and analyzed to find the best method for efficient recognition of diseases and pests of Cole plants. JL provided the facilities for data collection and contributed with the information for the data annotation. All authors read and approved the final manuscript.

Authors' information

Zhiyong Wei received his B.Sc. degree in 2005 from Guilin Institute of Technology, received his M.S. degree in 2016 from Guangxi University, now he is Senior Engineer in Nanning College For Vocational Technology.His main research interest include image and video processing algorithms and data mining.

Fengling Wang received the B.S received M.S. degree in 2006 from BeiJing University Of Technology. now he is Professor in Nanning College For Vocational Technology. His main research interests include computer software theory, image processing algorithms and data mining.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 22 January 2019 Accepted: 18 April 2019
Published online: 06 June 2019

References

1. Pereyra G, Tucker G, Chorowski J, et al. Regularizing neural networks by penalizing confident output distributions[J]. arXiv preprint arXiv:1701.06548, 2017
2. A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks[J]. Adv. Neural Inf. Proces. Syst. **25**(2), (2012)
3. O. Russakovsky, J. Deng, H. Su, et al., Imagenet large scale visual recognition challenge[J]. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
4. M. Everingham, L. Van Gool, C.K.I. Williams, et al., The pascal visual object classes (voc) challenge[J]. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
5. T.Y. Lin, M. Maire, S. Belongie, et al., in *European Conference on Computer Vision*. Microsoft coco: common objects in context[C] (Springer, Cham, 2014), pp. 740–755
6. J.R.R. Uijlings, K.E.A. Van De Sande, T. Gevers, et al., Selective search for object recognition[J]. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
7. S. Ren, K. He, R. Girshick, et al., Faster r-cnn: towards real-time object detection with region proposal networks[J]. IEEE Trans. Pattern Anal. Mach. Intell., **39**(6), 1137–1149 (2017)
8. W. Liu, D. Anguelov, D. Erhan, et al., in *European Conference on Computer Vision*. Ssd: single shot multibox detector[C] (Springer, Cham, 2016), pp. 21–37
9. O. Russakovsky, Y. Lin, K. Yu, et al., in *European Conference on Computer Vision*. Object-centric spatial pooling for image classification[C] (Springer, Berlin, Heidelberg, 2012), pp. 1–15
10. C. Szegedy, A. Toshev, D. Erhan, Deep neural networks for object detection[C]. Adv. Neural Inf. Proces. Syst., 2553–2561 (2013)
11. J. Dai, Y. Li, K. He, et al., R-fcn: object detection via region-based fully convolutional networks[C]. Adv. Neural Inf. Proces. Syst., 379–387 (2016)
12. J. Redmon, S. Divvala, R. Girshick, et al., in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. You only look once: unified, real-time object detection[C] (2016), pp. 779–788
13. J. Redmon, A. Farhadi, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. YOLO9000: better, faster, stronger[C] (2017), pp. 7263–7271
14. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014
15. K. He, X. Zhang, S. Ren, et al., in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Deep residual learning for image recognition[C] (2016), pp. 770–778
16. C. Szegedy, W. Liu, Y. Jia, et al., in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Going deeper with convolutions[C] (2015), pp. 1–9
17. S. Xie, R. Girshick, P. Dollár, et al., in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Aggregated residual transformations for deep neural networks[C] (2017), pp. 1492–1500
18. G. Huang, Z. Liu, L. Van Der Maaten, et al., in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Densely connected convolutional networks[C] (2017), pp. 4700–4708
19. Y. Chen, J. Li, H. Xiao, et al., Dual path networks[C]. Adv. Neural Inf. Proces. Syst., 4467–4475 (2017)
20. J. Hu, L. Shen, G. Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Squeeze-and-excitation networks[C] (2018), pp. 7132–7141
21. Zhang C, Bengio S, Hardt M, et al. Understanding deep learning requires rethinking generalization[J]. arXiv preprint arXiv:1611.03530, 2016
22. Bloice M D, Stocker C, Holzinger A. Augmentor: an image augmentation library for machine learning[J]. arXiv preprint arXiv:1708.04680, 2017
23. van Laarhoven T. L2 regularization versus batch and weight normalization[J]. arXiv preprint arXiv:1706.05350, 2017
24. N. Srivastava, G. Hinton, A. Krizhevsky, et al., Dropout: a simple way to prevent neural networks from overfitting[J]. J. Mach. Learn. Res **15**(1), 1929–1958 (2014)
25. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015

26. A. Fuentes, S. Yoon, S. Kim, et al., A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition[J]. Sensors **17**(9), 2022 (2017)
27. S.P. Mohanty, D.P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection[J]. Front. Plant Sci. **7**, 1419 (2016)
28. B. Liu, Y. Zhang, D.J. He, et al., Identification of apple leaf diseases based on deep convolutional neural networks[J]. Symmetry **10**(1), 11 (2017)
29. J. Amara, B. Bouaziz, A. Algergawy, A deep learning-based approach for banana leaf diseases classification[C]. BTW (Workshops)., 79–88 (2017)
30. S. Sankaran, A. Mishra, R. Ehsani, et al., A review of advanced techniques for detecting plant diseases[J]. Comput. Electron. Agric. **72**(1), 1–13 (2010)
31. Y. Kawasaki, H. Uga, S. Kagiwada, et al., *Basic study of automated diagnosis of viral plant diseases using convolutional neural networks[C]//International Symposium on Visual Computing* (Springer, Cham, 2015), pp. 638–645
32. S. Srdjan, A. Marko, A. Andras , et al., Deep neural networks based recognition of plant diseases by leaf image classification[J]. Comput. Intell. Neurosci. **2016**, 1–11 (2016)
33. K.P. Ferentinos, Deep learning models for plant disease detection and diagnosis[J]. Comput. Electron. Agric. **145**, 311–318 (2018)
34. Y. LeCun, C. Cortes, C.J.C. Burges, The MNIST database of handwritten digits, 1998[J]. URL http://yann.lecun.com/exdb/mnist. **10**, 34 (1998)
35. C. Wagner, *The sources of El cavallero Cifar[M]* (1903)
36. Abu-El-Haija S, Kothari N, Lee J, et al. Youtube-8m: a large-scale video classification benchmark[J]. arXiv preprint arXiv:1609.08675, 2016
37. A.W. Fitzgibbon, A. Zisserman, in *European Conference on Computer Vision*. Automatic camera recovery for closed or open image sequences[C] (Springer, Berlin, Heidelberg, 1998), pp. 311–326
38. A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm[C]. Adv. Neural Inf. Proces. Syst., 849–856 (2002)
39. P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation[J]. Int. J. Comput. Vis. **59**(2), 167–181 (2004)