Application of SSD framework model in detection of logs end

Hao Tang¹, Kejian Wang¹, Jiancai Gu², Xiaoye Li¹ and Wenhao Jian³

¹College of Information Science and Technology, Agriculture University of Hebei, Baoding 071000, China
²Forestry College, Agriculture University of Hebei, Baoding 071000, China
³Mulan paddock in Hebei Province state owned Forest Farm Administration Bureau, Chengde 067, Hebei, China

Abstract. External factors in the natural environment interfere with the detection of logs end. A method for detecting the log end using an SSD model in a natural scene is proposed. The use of default frame in SSD can realize the extraction and utilization of target features with different scales and improve the detection accuracy. By manually marking the foreground area of the log end, the interference of the features of the background area is reduced, and the convolution learning of the feature of the target area is enhanced. The performance evaluation and comparison test of the method showed that the accuracy rate was 94.87% and the recall rate was 91.34%. Compared with the traditional method, it reduces the occlusion effect caused by light and shooting angle, and improves the detection accuracy.

1. Introduction

Quickly and efficiently counting the amount of wood can increase industrial efficiency. Researchers have applied digital image recognition technology to detect fruits, insects, tubes, cells, and have achieved effective application value [1-7].

Logs can be classified as one of bars by their shape characteristics, and bar counts are usually detected with their end faces as targets. Therefore, it is feasible to carry out log counting by image processing means.

There have been many studies proposed in the literature related to the detection of the bars. These studies are mainly related to localization of the bars, which stands for detection of the location of the bars in the image.

Target extraction in RGB and LAB color space by clustering method [8-11]; target extraction using shape, texture and other features [12]; using Hough transform to get more evaluation indexes and SVM (support vector machine) to distinguish bar and background classification [13]. The watershed algorithm and the method based on concave segmentation are proposed to solve the problem of occlusion and conglutination in image processing [14,15]. These methods have achieved good results in the experimental environment. The experimental object of this paper is to identify the log end in the natural environment, so there are some interference factors, such as: light, shadow, stacking shelter, the size of wood is different from that of traditional bar with uniform specification, the above method is not ideal for the target detection in this case, and if the binocular vision is used, the cost will be increased[16].

Krizhevsky proposed Alex-net convolutional neural network [17], which has a good application in classification and recognition. Convolutional neural network has the ability of non-linear expression through activation function, and its generalization performance is improved by large training set, so it
can complete the task of object recognition and detection well. There are two kinds of target detection based on deep learning: (two stage) firstly, a series of candidate frames are generated by the algorithm as samples, and then the samples are classified by convolutional neural network, the representative networks are R-CNN, fast R-CNN, fasterR-CNN[18-20]; the other kind (one stage) regards the detection target position as a regression problem, and processes the whole image by using CNN network structure, the same as the typical networks include Yolo[21-23], SSD[24,25].

Logs end detection is a sub problem of target detection. The target detection model of convolution network does not need to design features manually. Through deep convolution network structure, the hidden features of image can be learned automatically to generate more reliable detection results. This paper constructs an SSD model structure and trains experimental images to obtain a better target detection model. This method can solve the problem of face detection and recognition under complex background conditions, and lays the foundation for quantitative statistics.

2. Materials
In this paper, the experimental material is to take the images of log stacking state in the forest farm of Chengde Forestry Bureau, and scale them to 300 × 300 through python script, so as to reduce the preprocessing pressure during network training. At the same time, data expansion and enhancement. Each image is expanded by rotation (180°, 270°), at the same time, we use LabelImage to mark the data manually to form XML file to make data set. In order to prevent over-fitting, the images are randomly scrambled and divided into training sets, verification sets and test sets to ensure the reliability of the later evaluation criteria.

In the marking process, considering the occlusion of the logs, the lack of light and the problem of not all the target areas, figure 1-3 selects the rectangular frame for all the end regions of the logs, and do not mark the section end with small areas. The XML file establishes the target attribute through annotation, which includes four labels to record the maximum and minimum values of coordinates x, y in the image, that is, the upper left corner and the lower right corner of the annotation box. It is used to determine the matching score according to the coincidence rate between the output value and the marked value when evaluating the model positioning function, and to establish the predefined output of the network model.

### Table 1. Data set structure and quantity

| Data Sets            | Test sets | Training sets | Verification sets | Total |
|----------------------|-----------|---------------|-------------------|-------|
| Logs end image       | 40        | 200           | 60                | 300   |
| expansion            | \         | 680           | 180               | 800   |

3. SSD model
SSD target detection model does not need candidate frame generation and repeated feature sampling. It can directly convolute the whole image and predict the categories and corresponding coordinates of the objects contained in the image, thus greatly improving the detection speed. Using small-scale convolution kernel and multi-scale prediction can improve the detection accuracy.

SSD target detection model does not need candidate frame generation and repeated feature
sampling. It can directly convolute the whole image and predict the categories and corresponding coordinates of the objects contained in the image, thus greatly improving the detection speed. Using small-scale convolution kernel and multi-scale prediction can improve the detection accuracy.

SSD network structure is divided into two parts: base network and auxiliary network. Basic network is also called backbone network, which usually uses the network with high-precision classification in the field of image classification (excluding its classification layer), such as VGG; Res-nets; Mobile networks. After the basic network, the convolution network structure of auxiliary network for target detection is linked, and the size of these auxiliary networks is gradually reduced so that multi-scale prediction can be carried out. Each auxiliary network layer will generate a feature map through a series of convolution kernels. Using the feature map, the main features in the original image can be preserved while the amount of calculation can be reduced. For a feature map (p is the number of channels, m and n are the size of the feature map), each auxiliary layer will use two $(3 \times 3 \times p)$ convolution kernels to predict and generate category prediction and position prediction.

The SSD model predicts k bounding boxes at each location of the feature map, predict the offset of the score and object position of a category target at this location relative to the label bounding box. Thus, $(c \times k)$ scores and $(4 \times k)$ position offsets are predicted at the position of each feature maps. For a feature map of size $m \times n$, a total of $(c + 4) \times k \times m \times n$ results will be predicted. Finally, the output result is subjected to non-maximum suppression to obtain a predicted value about the object type and position information in the image.

3.1 Building model

Figure 4 the detection model uses the first five scales of vgg16 as the base network model, performs convolution calculation through the front network, and outputs the third convolution layer under the fifth scale as the first feature map output (conv4_1). Then convolute the end full connection layer of the original vgg16 layer by layer.

Through convolution and average pooling, six characteristic layers, conv4_3, fc7, conv6_2, conv7_2, conv8_2, conv9_2, are obtained. Then two $3 \times 3$ convolution kernels are used to form the two sub feature maps of confidence and regression analysis. Because the smaller objects in the convolution process are not blurred in the previous several layers of calculation, usually the feature map with larger scale such as conv4_3 is used to detect the smaller objects; after several times of convolution, the smaller objects are filtered and the larger objects are retained, and the later feature maps such as conv9_2 are used to detect the larger objects.

![SSD model network structure diagram](image)

Figure 4. SSD model network structure diagram
Table 2. Main network structure diagram

| Layers               | Shape                                | Connected                                      |
|----------------------|--------------------------------------|------------------------------------------------|
| input_1 (InputLayer) | (None, 300, 300, 3) 0                | 0                                              |
| conv4_3 (Conv2D)     | (None, 38, 38, 512) 2359808           | conv4_3_norm_mbox_loc_reshape                  |
| fc7 (Conv2D)         | (None, 19, 19, 1024) 1049600          | fc7_mbox_loc_reshape                           |
| conv6_2 (Conv2D)     | (None, 10, 10, 512) 1180160           | conv6_2_mbox_loc_reshape                       |
| conv7_2 (Conv2D)     | (None, 5, 5, 256) 295168             | conv7_2_mbox_loc_reshape                       |
| conv8_2 (Conv2D)     | (None, 3, 3, 256) 295168             | conv8_2_mbox_loc_reshape                       |
| conv9_2 (avg pooling)| (None, 1, 1, 256) 295168             | conv9_2_mbox_loc_reshape                       |

3.2 Default boxes

For the sub-feature maps of the feature maps of six different scales, a default box (anchor frames) operation is performed. Match real tags with different scale default boxes. In order to adapt to the shape of the object to be detected, the length width ratio of the default boxes is set, and the corresponding number of different size feature maps is different. According to six kinds of feature graphs, the default boxes with different length width ratio [4,6,6,6,4,4] on each feature are respectively verified. The length width ratio of the default box is selected for different feature maps (Table 3). The specific length formula in the feature maps is given by the following equation:

\[
S_j = S_{min} + \frac{(S_{max} - S_{min})}{m-1} (j - 1) \quad j \in (1, m) \tag{1}
\]

Add side length \(S_k\) when aspect ratio is 1, formula is given by the following equation:

\[
S_j^a = \sqrt{(S_j S_{j+1})} \tag{2}
\]

\[
W_j^a = S_j \sqrt{a_r} \tag{3}
\]

\[
H_j^a = S_j / \sqrt{a_r} \tag{4}
\]

Where (formula 1) \(S_{min}\) and \(S_{max}\) are artificially set scales and the proportion of the size of the receptive field in the calculation. This paper sets 0.2 and 0.9 respectively. \(J\) represents the feature map number, so \(m = 6\). \(W_k^a\) and \(H_k^a\) are the width and height of the corresponding feature map respectively.

Figure 6 shows multiple default boxes (dashed frame) generated with \(a1\) and \(b1\) of two centers as the center under the size of 8 \(\times\) 8 feature map. Taking figure 7 as an example, \(c*k*m*n\) is the \(f\_confidence\) output, where \(C\) is the category in the classification \((c1, c2, \ldots, c_p)\), which represents the probability of each default boxes category. \(m\times n\) is the number of points under the feature map, i.e. size. The figure is 4 \(\times\) 4, 16 size. \(4*k*m*n\) is the output of \(f\_location\). 4 represents the coordinates of the center point and the width and height of each default box. \(K, m, n\) have the same meaning. At the same time, not every extracted default box is used for regression calculation, but the corresponding coincidence degree with the training set is extracted to form positive and negative samples. The threshold value of this paper is 0.5. Figure 8 and figure 9 show the case where different number of default boxes are marked on the detected image.
In the actual calculation, f_confidence is used for regression training, which is similar to the traditional classification convolution neural network. For f_location regression calculation, offset calculation is used to calculate the offset of the default box relative to the dimension coordinates and the width and height, formula is given by the following equation:

\[ l_{cx} = \frac{(b_{cx} - d_{cx})}{d_{w}} \]  
\[ l_{cy} = \frac{(b_{cy} - d_{cy})}{d_{h}} \]  
\[ l_{w} = \log\left(\frac{b_{w}}{d_{w}}\right) \]  
\[ l_{h} = \log\left(\frac{b_{h}}{d_{h}}\right) \]

The prior box position \(d = (d_{cx}, d_{cy}, d_{w}, d_{h})\) is marked in figure 5, and the corresponding generated boundary box \(b = (b_{cx}, b_{cy}, b_{w}, b_{h})\), \(l\) is the predicted value and the conversion value of both.

### 3.3 Optimizer and Loss function

The loss function (9) is defined as the weighted sum form of position error (location loss, loc) and confidence error (confidence loss, conf), in which the confidence error \(L_{\text{conf}}(x, c)\) uses the softmax loss function (10), and the position error \(L_{\text{loc}}(x, l, g)\) is the smooth L1 error function (11).

\[
L(x, c, l, g) = \frac{1}{N} \left( L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g) \right)
\]

\[
L_{\text{conf}}(x, c) = \sum_{i \in \text{Pos}} x_{ij}^{p} \log(\hat{c}_{j}^{p}) - \sum_{i \notin \text{Neg}} \log(\hat{c}_{j}^{p}) \quad \text{where} \quad \hat{c}_{j}^{p} = \frac{\exp(c_{i}^{p})}{\sum_{p} \exp(c_{i}^{p})}
\]

\[
L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} \text{smooth}_{\text{L1}}(l_{i}^{m} - \hat{g}_{j}^{m})
\]

Table 4. Loss function unit name

| Symbol | Significance | Symbol | Significance | Symbol | Significance |
|--------|--------------|--------|--------------|--------|--------------|
| \(N\)  | The number of matching default | \(x_{ij}^{p}\) | If the prediction box matches the real box, its the de | \(c\) | classification |
bounding box s. When n is 0, set l to 0.

fault box, j is the real box, and P is the classification label.

\(\alpha\) The weighting coefficient of position prediction and classification prediction error.

\(\text{smooth}_{L1}\) Smooth L1 error function

\(l_i^m\) The predicted bounding box coordinates, which are applied here (Equation 5-8) for conversion calculation.

\(g_j^m\) The bounding box coordinates of the label, here apply (Equation 5-8) for conversion calculation.

\(L_{\text{conf}}\) Softmax multi-classification error function

\(L_{\text{loc}}\) Position prediction error, using a smooth L1 error function

\(x_i^p \log(c_j^p)\) The prediction box matches the real box j with respect to the category p, and the higher the probability prediction of p, the smaller the loss.

\(\log(c_j^p)\) If there is no object in the prediction box, the higher the probability of predicting the background, the smaller the loss.

\(c_j^p\) Softmax classification

---

**Pos** Positive example  
**Neg** negative example

---

4. **Test results and analysis**

Using 16GB memory, Nvidia rtx2070 GPU, AMD 2700x CPU as hardware platform, windows 10 professional operating system, Cuda10.0 parallel computing framework, Cudnn v10.0 deep neural network acceleration library, using python programming language in Keras, Tensorflow deep learning framework to achieve the SSD target detection model and complete the training and verification of the model. The detection accuracy of the trained model is verified in the test set. Input the test set image into the trained network, detect the end face position of the log and record the detection results. Calculate the model prediction results and the marking box through the intersection and parallel ratio. If the threshold value exceeds 50%, the prediction is accurate. At the same time, accuracy (P) and recall (R) are used to measure and evaluate the detection accuracy, formula is given by the following equation:

\[
P = \frac{T_p}{(T_p + F_p)} \times 100\% \tag{12}
\]

\[
R = \frac{T_p}{(T_p + F_N)} \times 100\% \tag{13}
\]

\(T_p\) correctly detected the number of logs end, \(F_p\) mistakenly detected the non-logs end target as the number of logs end targets, and \(F_N\) mistakenly detected the logs end as the number of backgrounds.

In the results of 100 experiments, the accuracy of the model after training for the training set reached an average of 94.87%, and the recall rate was 91.34%.

Figure 10-12 shows some experimental results. Figure 13-15 compares and contrasts the traditional pre-processed image segmentation.


### Table 5. Experimental recognition rate

| sequence | $T_p$/quantity | $F_p$/quantity | $F_N$/quantity | SSD Precision(%) | SSD Recall(%) | Traditional Precision(%) | Traditional Recall(%) |
|----------|----------------|----------------|----------------|------------------|--------------|--------------------------|---------------------|
| 1        | 35             | 2              | 2              | 94.59            | 94.59        | 93.54                    | 89.36               |
| 2        | 25             | 0              | 1              | 100              | 96.15        | 89.24                    | 92.0                |
| 3        | 42             | 4              | 6              | 91.3             | 87.5         | 89.32                    | 92.59               |
| 4        | 45             | 4              | 3              | 91.83            | 93.75        | 87.93                    | 93.33               |
| 5        | 50             | 3              | 7              | 94.33            | 87.71        | 91.93                    | 91.67               |
| 6        | 75             | 8              | 12             | 90.36            | 86.2         | 93.54                    | 87.87               |
| 7        | 7              | 5              | 4              | 93.9             | 95.0         | 89.24                    | 87.36               |
| 8        | 65             | 7              | 7              | 90.27            | 90.27        | 89.88                    | 82.88               |
| 9        | 33             | 2              | 3              | 94.29            | 91.69        | 92.13                    | 85.0                |
| 10       | 57             | 5              | 10             | 91.93            | 85.07        | 93.20                    | 89.06               |
| ……       | ……             | ……             | ……             | ……              | ……          | ……                      | ……                 |
| 100      | 34             | 1              | 1              | 96.56            | 96.56        | 93.58                    | 86.05               |
| total    | 3629           | 196            | 344            | 94.87            | 91.34        | 91.88                    | 88.17               |

![Figure 10. SSD experiment 1](image)

![Figure 11. SSD experiment 2](image)

![Figure 12. SSD experiment 3](image)

![Figure 13. Traditional experiment 1](image)

![Figure 14. Traditional experiment 2](image)

![Figure 15. Traditional experiment 3](image)

SSD experiment 1-3 analysis, in which the error of accuracy is mainly due to the interference of some external factors, as well as the different features on some log sections (for example, the color of the section is darker due to wetting and the surrounding background fusion), the reason of recall is that there are many error factors, such as occlusion, surface texture and section shadow due to light. However, through contrast experiment 1-3, convolution training by labeling can effectively avoid the interference of external environmental factors, such as snow cover in contrast experiment 1 and small non target log cross section in the background, which will affect the cross section judgment during the image preprocessing process (all gray-scale classification is white, which will affect the later target segmentation); through contrast experiment 2, you can observe the shooting In contrast experiment 3, SSD training model is used to detect overlapped logs. In traditional methods, due to the few selected features, the overlapped logs are partially adhered, which will lead to the later target segmentation difficulty Hard. Therefore, compared with traditional machine vision, which relies on texture color and other features, the number of convolution layers is increased to expand the types of features, so as to avoid the interference of too few features, reduce the threshold setting operation due to different environments, improve the generalization ability in complex environments, and improve the detection accuracy.
5. Conclusions
Using SSD frame model for training, the obtained model can effectively detect and identify the logs end in the natural environment under the interference of log overlap, external debris and log cross section itself.

At the same time, to be improved: for the future improvement of accuracy, three solutions are proposed: (1) Improve the labeling specification and reduce the interference in the image pre-processing of the target area; (2) Collect more section data, expand the data set to improve the accuracy. For the improvement of recall rate, collect the existing factors affecting recall rate, such as occlusion, and expand the application scope of data set promotion model for these factors; (3) Improve network model and improve model efficiency.

References
[1] Cai Q, Chen SZ, Zhao R. (2015) Interaction relationship between central forestry investment and forestry economic growth in China[J]. Scientia SilvaeSinicae51(09):126-133.
[2] Diao G. (2014) The Study of Chinese timber supply and Policy[D], Beijing Forestry University, Beijing.
[3] J.R.Parker. (translater)Jing Li. (2012) Algorithms for Image Processing and Computer Vision. Tsinghua University Press, Beijing.
[4] ZHangWH, Zhu YJ. (2018) A Study on the Method of Counting Number of Ripe Apple Fruits Based on the Sample Obtained from Chinese Loess Plateau[j]. 2018,40(01):27-34. Journal of Southwest University (Natural Science Edition)
[5] Sumel. (2013) Estimation of mango crop yield using image analysis-Segmentation method [J].Computers and Electronics in Agriculture industry,91(57-64).
[6] CaoZH, MuSM, SunXX, SuTT. (2018) Research and Implementation of Sticky Board Pest Counting System Based on Android[J]. Journal of Henan Agricultural Sciences, 47(10):154-159.
[7] Gao P, QianYS, WangPL, LuX. (2018) The Method of Cotton Leaf Segmentation under the Complex Background in Cotton Fields[J]. Xinjiang Agricultural Sciences. 55(12):2288-2295.
[8] Li ZR, (2017) Design and implementation of the bar recognition and counting system based on image processing[D]. Zhengzhou University, Henan.
[9] Li L, (2017) Research of counting system for oil casing based on image processing[D]. Taiyuan University of Technology. Shanxi.
[10] Liu BL, (2015) Research and Application of Binding Bar Counting Method Based on Machine Vision. Tiangong University, Tianjin.
[11] ZhongXX, JingL, LinYH. (2017) Log-end Area Recognition Based on Lab Color Space and Threshold Segmentation[J]. 2017,35(02):95-99.
[12] MEI ZR, REN HE, ZHU M. (2012) Algorithm of log end recognition based on non-linear least squares principle[J]. Com-puter Engineering and Applications, 48 (2) : 177-178.
[13] Mao QZ, Pan ZM, Gao WW. (2014) Using the iterative Hough circle transformation to achieve reliable counting of bundled bars[J] Geomatics and Information Science of Wuhan University, 39(03):373-378.
[14] Huang L, Qi BQ, Hu B. (2012) Automatic Counting Method of Steel Bar in a Bundle Based on Connected Domain Area of Image[J]. Industry and Mine Automation,38(05):77-79.
[15] Lin YH, Jing L, Wang CY, Wang XP, LinJX, Lin PJ. (2016) Outline identification and verification of cross section of log based on arc edges[J]. Journal of Fujian Agriculture and Forestry University (Natural Science Edition). 45(06):649-654.
[16] Chen GH, Zhang Q, Chen MQ, LI JW, YIN HY. (2018) Rapid detection algorithms for log diameter classes based on binocular vision[J]. Journal of Beijing Jiaotong University. 42(02):22-30.
[17] Krizhevsky A, Sutskever I, Hinton G. (2012) ImageNet Classification with Deep Convolutional
[18] GIRSHICK R, DONAHUE J, DARRELL T, et al. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation [C]In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colombia 580 − 587.
[19] GIRSHICK R. (2015) Fast R-CNN [C]In: Proceedings of the IEEE International Conference on Computer Vision. Santiago de Chile. 1440 − 1448.
[20] Ren S, He K, Girshick R, et al. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 39(6):1137-1149.
[21]REDMON J, DIVVALA S, GIRSHICK R, et al. (2016) You only look once: unified, real-time object detection[C]In: Computer Vision and Pattern Recognition. IEEE, Las Vegas 779 − 788.
[22] REDMON J, FARHADI A. (2017) YOLO9000: better, faster, stronger [C][In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, Hawaii State.6517 -6525.
[23] XiongJT, Liu Z, Lin R, ChenSM, ChenWJ, YangZG. (2018) Unmanned Aerial Vehicle Vision Detection Technology of Green Mango on Tree in Natural Environment
[24] Peng HX, Huang B, Shao YY, Li ZS, Zhang CW, Chen Y, Xiong JT. (2018) General improved SSD model for picking object recognition of multiple fruits in natural environment[J]. Transactions of the Chinese Society of Agricultural Engineering. 34(16):155-162.
[25] Zhang S, Benenson R, Omran M, et al. (2017) Towards Reaching Human Performance in Pedestrian Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, PP (99):1-1.