Structured Light Detection Algorithm based on Deep Learning

Structured Light Detection Algorithm based on Deep Learning

Shujie Ma[a], Yunbo Song, Na Cheng, Yun Hao, Zhengyu Chen and Xianping Fu[*]

School of Dalian Maritime University, Dalian 116000, China

*Corresponding author e-mail: fxp@dlmu.edu.cn, [a]562269866@qq.com

Abstract. This paper proposes a network that applies deep learning to structured light detection. With the powerful learning ability of the artificial neural network, the image can be effectively processed in an end-to-end way after simple processing of the image.

1. Introduction

Structured light is widely used in industrial detection. Combined with the breakthrough in the field of artificial neural networks in recent years, this paper proposes a SL-Rcn (Structured light Rcnn) which is based on Faster-Rcnn [13] and suitable for structural light detection. The network first divides the image of a more complex object into multiple independent parts, and then extracts the structured light strips of each part, which are then employed by the SL-Rcnn network to obtain the need information. Last, the entire object image is obtained by merging. Extracting structured light stripe can effectively shield unnecessary features and environmental factors, and segmenting the image into independent components can better detect the characteristics of the local area. The above methods have achieved good results in practice. For example, in the field of industrial cutting of irregular objects (such as industrialized cutting of marine products), local cutting points need to be determined. Experiments show that mAP is 95.3% in our proposed method when existing a lot of noise and the object can be detected in real time.

2. Related work

2.1. Object Proposals

This is an area that has been studied in many fileds. The tracking and investigation of related content can be found in [7-9]. In a broad sense, the Object Proposals method includes algorithms such as Selective Search [3] based on grouping super-pixels and algorithms such as EdgeBoxes [6] based on sliding windows. Object Proposals has been considered a separate part independent of detection modules in a long time, such as Selective Search [3] object detectors, R-CNN [4], and Fast R-CNN [2].

2.2. Deep Networks for Object Detection

The R-Cnn [4] network trains the CNN model by end-to-end, thus classifying the foreground and background of the propsal regions. Its accuracy heavily depends on the performance of the region proposal model [8]. For the prediction of the bounding-boxes of the object, there are also some papers that propose different methods to predict the bounding boxes of objects using deep neural networks

[10], [6], [11], [12]. The shared computing of convolutional, because of the good computational performance of the computer vision field [6], [1], [5], [2], Fast R-CNN [2] on the network of convolution shared computing with good accuracy and speed is also trained.

3. SL-Rcnn

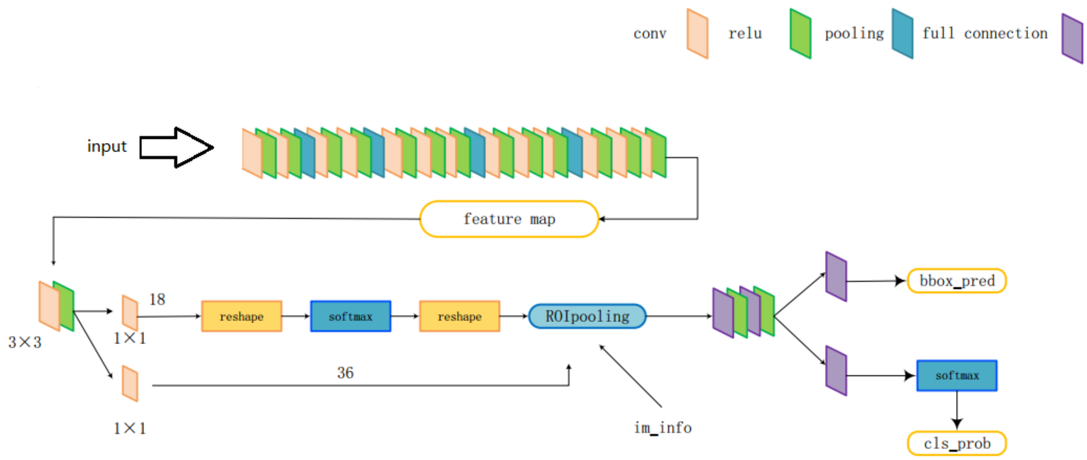

Figure 1. The main framework of RL-Rcnn: the shared convolutional layer is generated by the multi-layer convolutional neural network above.r, and then the feature map enters the RPN network for proposed regions extraction and calculation.

First, we convert the RGB image into an YCbCr image. In YCbCr color space, Y represents brightness, Cb represents blue chrominance component, and Cr represents red chrominance componentm, and it has the advantages of luminance and chrominance separation. The target light can be easily obtained by extracting the Cr component stripe. We split an original picture into k parts (k = 10 in one of our application scenarios), the longer k is, the longer it takes, the more each is illuminated by structured light, and the structured light is acquired through the YCbCr color space. Stripe, then use the resulting image as the input to SL-Rcnn.

Our network consists of multiple parts: the first part is mainly a deep fully convolutional network, which obtains the high-dimensional features of the structured light image after multi-dimensional convolution of the image. The second part is based on the Fast R-CNN detection network [2], the input is the proposed regions extracted on the high-dimensional features of the first part of the output. The third part is an improved network based on the Region Proposal Network (PRN) [13], which aims to tell Fast R-CNN how the network is concerned with the features extracted by deep convolutional networks.

3.1. Optimized Region Proposal Network

The RPN network [13] can receive an input of any size from the convolutional layer obtained in the first part, and output a series of rectangular object proposals with scores. We use the convolutional neural network to accomplish this task [5], and the output of proposals is followed. To detect the input of the network, the RPN network consists of two branches, the first branch is used to classify the foreground and the background, and the second branch is used to do the bouding box regression to get a rough object border, here the RPN network and Fast - The Rcnn detection network [2] shares a set of identical convolutional layers, as shown in Figure 2.

To get the region proposals, we slide a mini-network on the last layer of the shared convolutional layer, which is part of the RPN network. The mini-network runs on the convolutional layer by sliding the window. Therefore, the following network will share parameters for all inputs. On the one hand, the total amount of parameters is reduced, the calculation speed is accelerated, and the model is reduced. The risk of integration increases the generalization ability. In order to speed up the operation of the network without significantly affecting the IOU, we have given the mini-network a step of size 2, which effectively reduces the amount of computation.

The RPN network performs convolution calculation on the shared convolutional layer through a common n*n convolutional network, and then respectively sets a 1*1 convolutional layer for the foreground and background classification networks and the bounding box regression network. Cross-channel information integration and the role of adjusting feature dimensions.

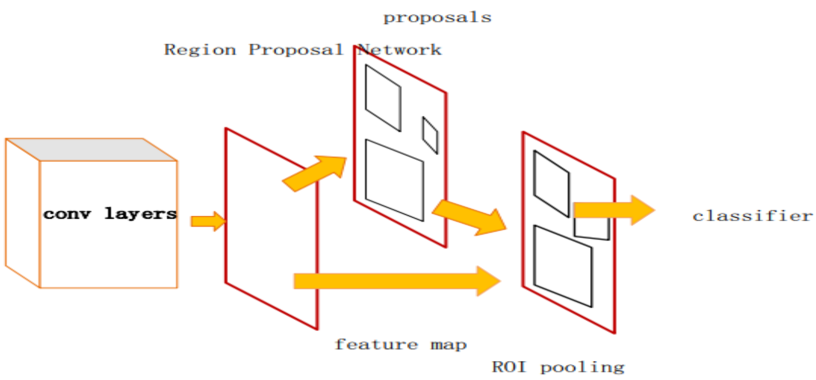

Figure 2. RPN-SubNetWork.

3.2. Optimized Anchors

In the process of calculating proposed regions in the RPN network [13], mini-network performs convolution calculation in the form of sliding window [13]. For each position, we will predict multiple regions proposals at the same time. We will maximize the number of each position. The proposal is marked as k. Since each region proposals is determined by four points, each position, the bounding box regression network reg will output up to 4k numbers, and the similar classification network cls will output 2k scores representing the probability scores of each proposal containing objects. Does not contain the probability score of the object.

Anchors are located in the center of the sliding window [13] and are closely related to the length and width ratio of the generated proposals. The original Faster-Rcnn uses the default nine aspect ratio anchors to approximate the aspect ratio of all objects. Due to the particularity of the structured light, we optimized all nine aspect ratios to improve detection accuracy. Optimize the original ratio to 15:1, 10:1, 10:3, 10:4 and the corresponding 1:15, 1:10, 3:10, 4:10 and 1:1 separately. For example, an anchor with an aspect ratio of 15:1 is very effective for long and narrow strip images under single-line structured light detection. For a convolutional feature map with width W and height H, there are W*H*k/2 anchors.

In addition, the anchors mechanism also brings the advantage of translation invariance for structured light detection. Since the structured light will cause different degrees of jitter during the working process due to objective reasons such as mechanical vibration, the light strip will inevitably produce some unstable changes. When extracting region proposals in the RPN network [13], the corresponding features are extracted by the anchors [13] shared convolution layer. If the structural light position is shifted due to mechanical jitter, etc., it is used to calculate the features extracted by the anchors. The network is shared by all input parameters, so even if the location changes, the anchors and parameter sharing mechanisms ensure that the structured light bars at different locations are still accurately identified.

3.3. Loss Function

The improved anchors mentioned above extract different features. Based on the application scenarios and data, we find the appropriate optimization parameters: (i) For each ground-truth box, the anchor with the highest overlap is directly determined as a positive sample [13]. (ii) Compared to large and diverse datasets, our network is used to detect specific content, so the anchor/anchors with an IOU value greater than 0.88 are positive for any ground-truth box. In Faster-RCNN network, this value is 0.7. Experiments have shown that in the current scenario, selecting anchor/anchors with an IOU value greater than 0.88 can effectively improve the quality of the positive samples without causing significant side effects.

Based on the above convention, our loss function for each picture is as follows:

$$L\{(p_i)'(t_i)\} = \frac{1}{Ncls}\sum_i L_{cls}(p_i, p_j^*) + \lambda \frac{1}{Nreg}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{1}$$

In the above formula, the subscript i represents each mini-batch and p_i represents the probability value of each anchor as an object. If the anchors are judged to be positive samples, the value is 1, otherwise the value is 0.t_i^* is a vector representing the four parameterized coordinates of the bounding box. A parameterized coordinate vector representing the four bounding boxes associated with the anchors that are determined to be positive samples. $Ncls$ is a binary classification loss function that classifies positive and negative samples, and log loss is utilized in this paper. For regression losses, we use the same loss function as defined in Fast-RCNN. The expression $p_i^* L_{reg}$ in the formula is valid only when the value p_j^* is 1, otherwise the value is invalid. As the balance parameter, λ plays a role in balancing weights.

4. Experiment And Conclusion

We collect datasets (hereinafter referred to as datasets) collected in the actual environment, including 10,000 images of the training set and 2000 images of the test set. The training set includes 6,000 samples containing the detected object and 4000 samples containing no detected object, and the test set contains 1000 samples of the test object and the sample containing no test object. In the experiment, the convolution network uses Image-Net's pre-train network VGG-16 model, which includes 13 convolutional layers and 3 fully connected layers.

We compared the Fast-Rcnn [2], YOLO, Faster-Rcnn [13] and RL-Rcnn in this paper on the dataset, and obtained the accuracy of different networks for detecting samples containing detected objects (Positive Accuracy). The accuracy of the detection object (Negative Accuracy) and the accuracy of the detection frame on the object are expressed by IOU (Intersection over Union), and finally the time (ms) required for calculation of each picture and different models is given.

Table 1. Results.

Model	Positive	Negative	IOU	Time(ms)
Fast-Rcnn	79.3	76.2	0.78	830
YOLO	80.5	83.6	0.74	57
Faster-Rcnn	91.8	93.5	0.86	192
RL-Rcnn	**96.0**	**94.5**	**0.91**	**155**

5. References

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, in European Conference on Computer Vision (ECCV), 2014.

[2] R. Girshick, “Fast R-CNN”, in IEEE International Conference on Computer Vision (ICCV), 2015.

[3] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition”, International Journal of Computer Vision (IJCV), 2013.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[6] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks", in International Conference on Learning Representations (ICLR), 2014.
[7] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" in British Machine Vision Conference (BMVC), 2014.
[8] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What makes ´ for effective detection proposals?" IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2015.
[9] N. Chavali, H. Agrawal, A. Mahendru, and D. Batra, "Object-Proposal Evaluation Protocol is 'Gameable'", arXiv:1505.05836, 2015.
[10] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection", in Neural Information Processing Systems (NIPS), 2013.
[11] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks", in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[12] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection", arXiv:1412.1441 (v1), 2015.
[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015.