C-CNN: Cascaded Convolutional Neural Network for Small Deformable and Low Contrast Object Localization

Xiaojun Wu[1,2(✉)], Xiaohao Chen[1], and Jinghui Zhou[1]

[1] School of Mechanical Engineering and Automation, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, Guangdong, China
wuxj@hit.edu.cn

[2] Shenzhen Key Laboratory for Advanced Motion Control and Modern Automation Equipment, Shenzhen 518055, Guangdong, China
http://smea.hitsz.edu.cn/Userfiles/wuxiaojun/

Abstract. Traditionally, the normalized cross correlation (NCC) based or shape based template matching methods are utilized in machine vision to locate an object for a robot pick and place or other automatic equipment. For stability, well-designed LED lighting must be mounted to uniform and stabilize lighting condition. Even so, these algorithms are not robust to detect the small, blurred, or large deformed target in industrial environment. In this paper, we propose a convolutional neural network (CNN) based object localization method, called C-CNN: cascaded convolutional neural network, to overcome the disadvantages of the conventional methods. Our C-CNN method first applies a shallow CNN densely scanning the whole image, most of the background regions are rejected by the network. Then two CNNs are adopted to further evaluate the passed windows and the windows around. A relatively deep model net-4 is applied to adjust the passed windows at last and the adjusted windows are regarded as final positions. The experimental results show that our method can achieve real time detection at the rate of 14FPS and be robust with a small size of training data. The detection accuracy is much higher than traditional methods and state-of-the-art methods.

Keywords: Deep learning · Convolution neural network
Small deformable and low contrast detection

1 Introduction

Object localization is an important task in the process of industrial automatic production, for example, pick and place of an industrial robot, position localization in surface mount technology (SMT) etc. Template matching method is

X. Wu—This work was supported by the Basic Research Key Project of Shenzhen Science and Technology Plan (No. JCYJ20150928162432701, JCYJ2017041315253 5587, JCYJ20170307151848226).

J. Yang et al. (Eds.): CCCV 2017, Part I, CCIS 771, pp. 14–24, 2017.
https://doi.org/10.1007/978-981-10-7299-4_2

mostly adopted for this task in machine vision applications. NCC based template matching applies normalized cross correlation as the features [1], shape based template matching adopts edge feature for object localization [2]. In many cases, this kind of methods performs well, but these methods have drawbacks in that they are less robust against geometrical distortion including occlusion, deformation, illumination distortion, motion blur or extreme low contrast.

In recent years, CNN based object detection methods have achieved state-of-the-art result on classification tasks [3–5] and object detection tasks [6–8]. Current CNN based object detection methods mainly focus on general object detection in nature sense. As objects have different scales and different aspect ratios, sliding window based method [9–12] will lead to a very large computation complexity, so region proposal based and regression based methods are adopted. This kind of method was first proposed by Girshick et al. [6]. For accelerating, Fast-RCNN [13] by Girshick and Faster-RCNN [8] by Shaoqing Ren were proposed. Recently, regression models based methods became a new research hotspot [14,15] and more rapid methods, like YOLO [16] and single shot multibox detector (SSD) [17], are proposed.

There are 3 main challenges for applying current CNN techniques to object localization in the industrial environments, (1) it's unrealistic to label a lot of training data for a single scene in industry; (2) the method should be fast enough to cope with the large-capacity production line; (3) it should be robust to deal with the variety of products. Current state-of-the-art methods, like SSD [7] and Faster-RCNN [8], are rapid, but these methods require a huge amount of training data and training time, which is not suitable for practical applications in industry.

In the industrial automatic inspections, objects usually have a fixed scale and aspect ratio, so we only need to scan the whole image by only one fixed window, which makes the computation complexity acceptable. What's more, this kind of methods have less false negative. But we find if only adoption a signal model, there are many false alarms in the localization results.

Above all, we design a cascaded convolutional neural network (C-CNN) based method for object localization in the industrial environment. The proposed C-CNN can achieve a rapid localization speed and it is robust even we only use a small size of training data. Our method runs 14 FPS on GTX970. In the following section, we present the overall framework of proposed method and the details, and then we introduce our experiments and compare our method to traditional template matching and current state-of-the-art CNN based methods.

2 Cascaded CNN Detector

For object localization in industry, our object detector is shown in Fig. 1. Given a testing image, we first resize the image to a small scale and use net-16 to densely scan the whole image to reject most of the background windows. Then two networks, net-32-1 and net-32-2, further reject the remained background windows. The passed windows are accepted as the rough detection results. We apply a

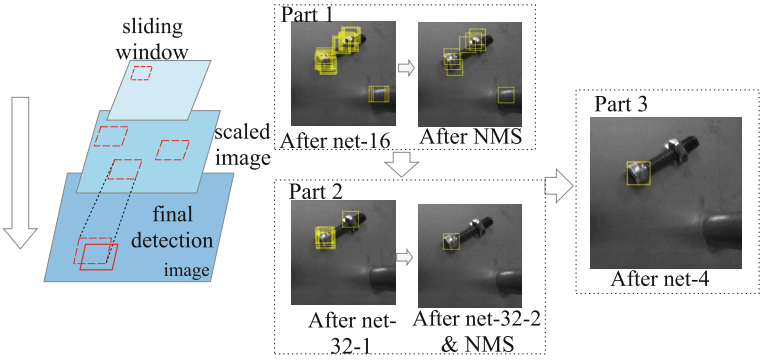


Fig. 1. Testing pipeline of our method. First we detect in the scaled image and adjust in the original image finally.

relatively deep model net-4 to adjust the passed windows. Non-maximum suppression [17] (NMS) is adopted to eliminate highly overlapped detection windows at the end of part 1 and part 2, seeing in Fig. 1.

2.1 The net-16

Training of the net-16. The structure of net-16 is shown as Fig. 2, which is a binary CNN classifier for classifying objects and backgrounds. We select Rectified Linear Units (ReLU) as our active function [18]. ReLU has been widely used in many work [3], and has been proved that it can improve expression ability of network and speed up convergence. Softmax loss function is adopted as our cost function; we also apply weight decay for avoiding overfitting.

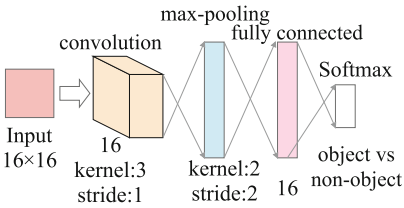


Fig. 2. CNN structures of net-16.

We crop the object patches as the positive examples, and other regions have an intersection over union (IOU) less than 40% with objects are regarded as negative examples. The numbers of negative examples are much more than positive examples. So we adopt rotating, Gaussian blur and Gaussian noise for data augmentation. We keep the positive and negative examples have a ratio $1 : 4$ while training.

Testing of the net-16. We adapt fully convolutional network instead of densely scanning the whole image, which can eliminate redundancy computation and achieve same results. When testing, the fully connected layer of net-16 is converted to a convolutional layer with 7×7 kernel size.

For a $W \times H$ testing image with $a \times a$, we first resize the image to $w \times h$. Here, $w = 16/W$, $h = 16/H$. Then input the resized image into the converted net-16 and obtain a map of confidence. Every confidence refers to one window on the original image, and the windows with a confidence less than threshold $T_1 = 0.9$ will be rejected.

2.2 The net-32

The net-32 is divided into two sub-network net-32-1 and net-32-2. Both networks are binary classifiers.

Training of the net-32. Through our experiments, using only one network cannot reject fault detection, so we design a deep network net-32-1 and a shallow network net-32-2. The two network structures are shown as Fig. 3. Deep network structure will help to extract more semantic information, and the shallow structure can retain more details [19].

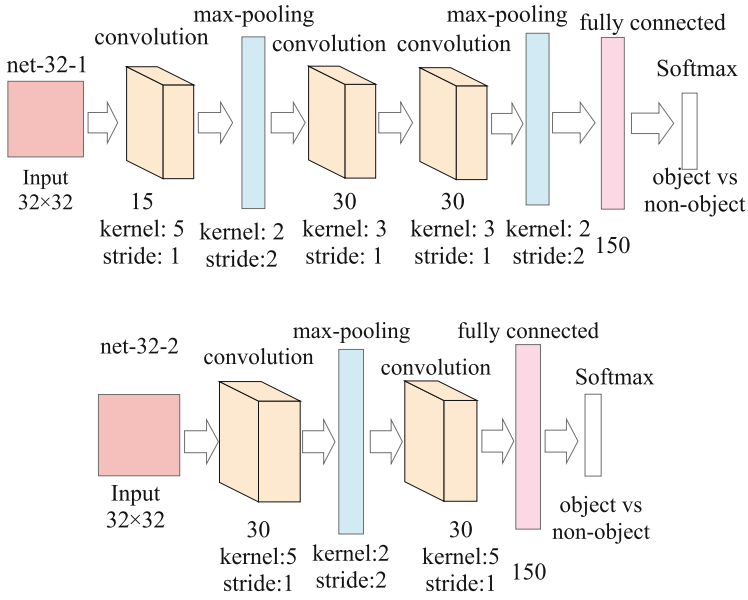


Fig. 3. CNN structures of the net-32-1 and net-32-2.

For the training of these two networks, we apply the trained net-16 to carry out hard negative mining. We use net-16 to scan the images, the windows which

have confidence higher than T_1 and have an IOU less than 0.5 with positive windows will be regard as negative examples.

Considering the remaining windows of net-16 will be not accurate enough. We randomly sample windows which have same size with objects and have more than 70% IOU with ground truth as positive examples. We also employ data augmentation strategy in this part.

Testing of the net-32. We evaluate the passed windows of net-16 and the 8 windows around them by net-32. Given a passed window (x, y, a, a) centering at (x, y) of (a, a) size and the size of original image is $W \times H$. Then our evaluated windows are (x', y', a, a), where $x' = x \pm (rS_x)$, $y' = y \pm (rS_y)$. Here, $r \in \{-0.75, 0, 0.75\}$, $S_x = 2W/16$ and $S_y = 2H/16$. The windows corresponded to a confidence higher than threshold $T_2 = 0.85$ are regarded as detection windows.

2.3 The net-4

The net-16 and net-32 are not accurate enough, so we train another CNN, called net-4, to adjust the detecting windows. The net-4 is a 9-class classifier for the object pattern and its eight surrounding patterns.

Training of the net-4. As object region patterns and its surrounding patterns are similar to each other, we design a relatively large CNN for this task. For an object of size $a \times a$, the input size of net-4 is $a \times a$, and the outputs of net-4 are the confidence corresponded to these 9 regions. The structure of net-4 is shown as Fig. 4. net-4 has a similar structure with net-32-1, but the structure of net-4 is wider than net-32-1 for obtaining more information.

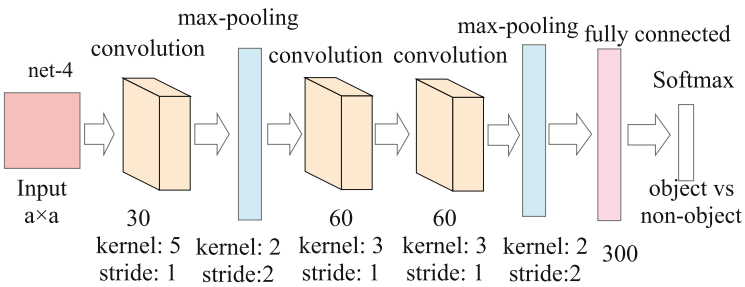


Fig. 4. net-4 consists of three convolution layers, two pooling layers, one fully connected layer and a softmax classifier.

For an object window (x, y, a, a), we crop the object windows' patches and 8 surrounding windows' patches as our training data. These nine windows can be expressed as

$$(x + r_x a, y + r_y a, a, a), \tag{1}$$

where $r_x \in \{-0.15, 0, 0.15\}$ and $r_y \in \{-0.15, 0, 0.15\}$. We also apply some data augmentation measures. It should be noted that the rotated patterns should be cropped from the rotated images instead of rotating the cropped patterns directly.

Testing of net-4. The net-4 accepts the passed windows as input and distinguishes which pattern of this window should be, then adjusts the center coordinate of the detection windows. The adjusting process is shown as Fig. 5.

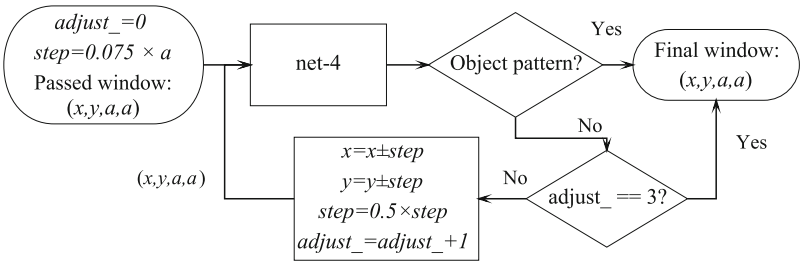


Fig. 5. The adjusting pipeline for a passed window.

3 Experiments and Analysis

In this section, we evaluate the performance of the proposed approach and compare our method to other methods. We adopt precision rate (P_{rate}) and recall rate (R_{rate}) to measure the performance of these algorithms. P_{rate} and R_{rate} are defined as Eq. (2).

$$\begin{cases} P_{rate} = \frac{TP}{TP+FP} \\ R_{rate} = \frac{TP}{TP+FN} \end{cases} \tag{2}$$

where TP is true positive; FP is false positive; FN is false negative.

3.1 Experiments Results

For evaluating our proposed algorithm, we test our method in different image sets. Each of image sets are compose of 100–200 images with resolution of 640 × 480. Figure 6 demonstrates some sample images from the testing image data sets. We can see that there are lots of interferences and the backgrounds are also very complex, and the targets are very small, blur and deformed, which make the detection task very difficult.

We only picked out and annotate about 20–30 images of them as the training data and the testing result is shown as Table 1. We find our method is robust although the training set is only composed of 20–30 images. All of P_{rate} and R_{rates} are higher than 97% in our experiments.

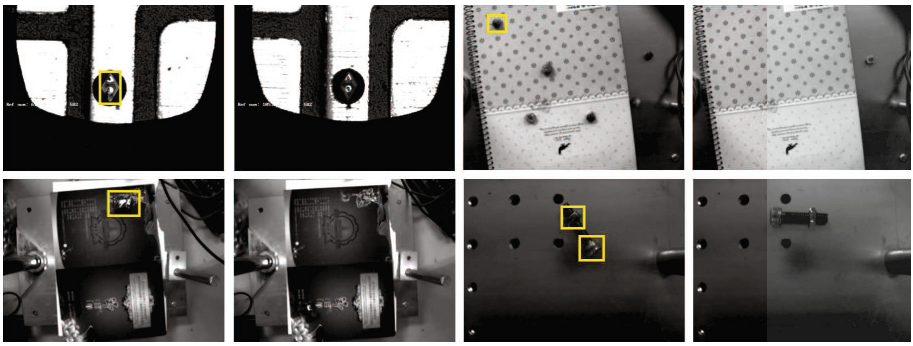


Fig. 6. Some sample images from testing image data sets, the top left called semi images, the top right called flower images, the bottom left called candy images, and the bottom right called screw images.

Table 1. Results of our proposed method

		Object	Background	P_{rate}	R_{rate}
Semi images	Positive	369	12	96.85%	98.66%
	Negative	5	0		
Flower images	Positive	208	0	100%	97.1%
	Negative	6	0		
Candy images	Positive	98	2	98.0%	98%
	Negative	2	0		
Screw images	Positive	279	3	98.93%	99.64%
	Negative	1	0		

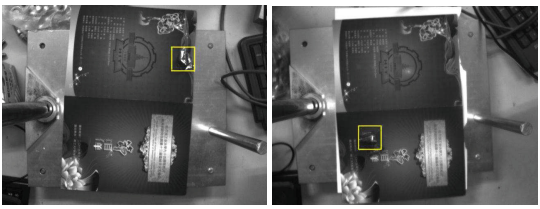
3.2 Comparisons Results and Analysis

We compare our method with the shape based template matching in Mvtech Halcon [21] (from a Germany based machine vision company) and SSD [7] method. For template matching algorithm, only one template image is needed. For SSD [7], We also pick out about 20–30 images as training data. We compare these three methods in different image sets. The results are similar, so we only show the result of candy images in Table 2. In the candy images, the candy wrapping paper is reflecting, deformable, distorted and low contrast with the background. The result is shown as Fig. 7.

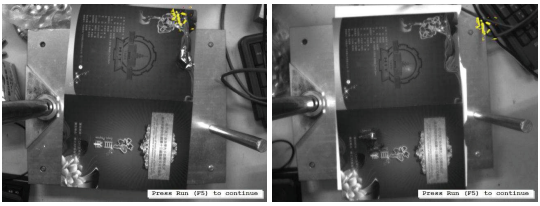
The results show that our method has a better performance than the template matching and SSD, the yellow box in Fig. 7. When the objects have large various visual changes and the shape features of objects are not obvious, the template matching nearly fails to detect the objects. Although we adopt 4-CNN models in our method, we apply multi-scale detecting and fully convolutional network to accelerate the algorithm. As we adopt the CNN in our detector, our method is easy to be parallelized on GPU. When using a moderate GPU card, GTX970, our

Table 2. Results of template matching and SSD

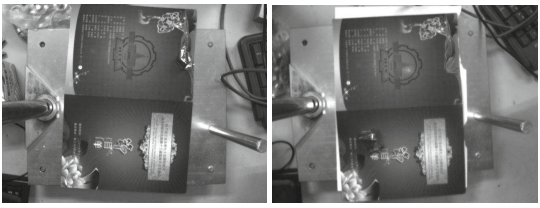
		Object	Background	P_{rate}	R_{rate}
Template matching	Positive	11	89	11%	11%
	Negative	89	0		
SSD	Positive	91	0	100%	91%
	Negative	9	0		



(a) Results from proposed method



(b) Results from template matching



(c) Results from SSD

Fig. 7. Object localization results of our proposed method (top), template matching (middle) and SSD (bottom). (Color figure online)

method can achieve about 14 FPS which is comparable to traditional template matching methods. The runtime comparison illustrates in Table 3.

We also compare our method with a commercial software ViDi Suite [22], a deep learning based industrial image analysis software, developed by a Swiss software firm to solve industrial vision challenges. We also provide 20–30 images as training data for ViDi. The result is shown in Table 4 and Fig. 8.

As the results shown in Table 4 and Fig. 8, the ViDi has a higher P_{rate}. But our method can obtain a better recall rate in three of these four image sets. Particularly, ViDi only has 75% R_{rate} in Candy images. As the background is

Table 3. Runtime comparison

Methods	Runtime for one image
Template matching (on CPU)	43 ms
SSD (on GTX970)	45 ms
Proposed method (on CPU)	633 ms
Proposed method (on GTX970)	72 ms

Table 4. Results of ViDi object localization method

		Object	Background	P_{rate}	R_{rate}
Semi images	Positive	369	0	100%	98.66%
	Negative	5	0		
Flower images	Positive	203	0	100%	95.75%
	Negative	25	0		
Candy images	Positive	75	0	100%	75%
	Negative	25	0		
Screw images	Positive	257	0	100%	91.79%
	Negative	23	0		

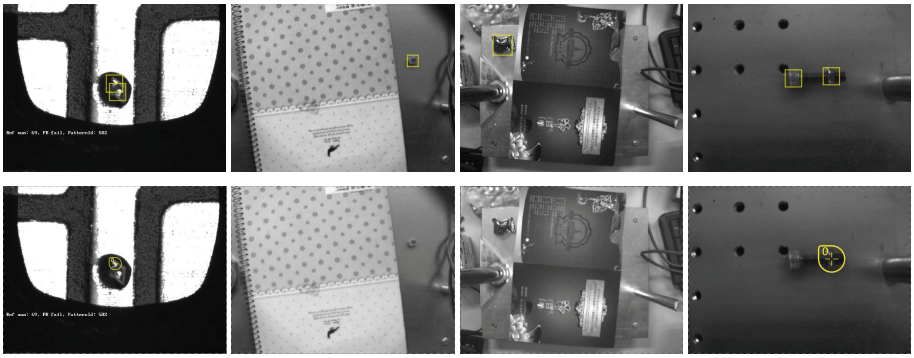


Fig. 8. The first row shows the results of proposed method and the second shows the localization results of ViDi.

usually stationary in industrial machine vision applications, the object localization is different from the wild object detection. There is not any publicly available dataset to compare.

4 Conclusion

Object localization is an important task in the industrial machine vision applications. Traditional template matching methods will completely fail to detect

objects in some extreme cases and current CNN based methods are focused on general object detection in nature sense. We propose a cascaded CNN detector, C-CNN, specifically for object detection in industrial sense. The C-CNN method is proved to be robust through our experiments and can locate the objects in extremely poor quality images. It can outperform the traditional methods and the state-of-the-art methods with small number of the training images. Furthermore, the real time performance of our method is achieved on a moderate GPU. It can be utilized in practical machine vision systems.

References

1. ShinIchi, S.: Simple low-dimensional features approximating NCC-based image matching. Pattern Recognit. Lett. **32**(14), 1902–1911 (2014)
2. Hou, Q.Y., Lu, L.H., Bian, C.J., Zhang, W.: Template matching and registration based on edge feature. In: Photonics Asia International Society for Optics and Photonics, pp. 1429–1435 (2013)
3. Alex, K., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Neural Information Processing Systems (NIPS), pp. 1097–1105 (2012)
4. Szegedy, C., Liu, W., Jia, Y.Q., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015)
5. He, K.M., Zhang, X.Y., Ren, S.Q., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
6. Ross, G., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587 (2014)
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot multibox detector. arXiv preprint arXiv:1512.02325 (2015)
8. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1 (2016)
9. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Yann, L.C.: Overfeat: integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
10. Li, H.X., Lin, Z., Shen, X.H., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5325–5334 (2015)
11. Farfade, S.S., Saberian, M.J., Li, L.J.: Multi-view face detection using deep convolutional neural networks. In: International Conference on Multimedia Retrieval ACM, pp. 224–229 (2015)
12. Chen, X.Y., Xiang, S.M., Liu, C.L., Pan, C.-H.: Vehicle detection in satellite images by parallel deep convolutional neural networks. In: Asian Conference on Pattern Recognition (IAPR), pp. 181–185 (2013)
13. Girshick, R.: Fast R-CNN. In: IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015)
14. Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Neural Information Processing Systems (NIPS), pp. 2553–2561 (2013)

15. Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3476–3483 (2013)
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. arXiv preprint arXiv:1506.02640 (2015)
17. Adam, H., Hradi, M., Zemk, P.: EnMS: early non-maxima suppression. Pattern Anal. Appl. **15**(2), 121–132 (2012)
18. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. Aistats **15**(106), 275–283 (2011)
19. Wanli, O., et al.: Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. arXiv preprint arXiv:1409.3505 (2014)
20. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
21. Mvtech Halcon. http://www.mvtec.com/products/halcon/
22. Vidi suite. https://www.vidi-systems.com/

http://www.springer.com/978-981-10-7298-7

Computer Vision
Second CCF Chinese Conference, CCCV 2017, Tianjin, China, October 11–14, 2017, Proceedings, Part I
Yang, J.; Hu, Q.; Cheng, M.; Wang, L.; Liu, Q.; Bai, X.; Meng, D. (Eds.)
2017, XXIV, 771 p. 373 illus., Softcover
ISBN: 978-981-10-7298-7