



An Industrial Defect Detection Platform Based on Rapid Iteration

Jianchao Zhu¹(✉), Dong Cheng¹, and Qingjie Kong²

¹ East China Normal University, Shanghai, China
51174500068@stu.ecnu.edu.cn, dong_cheng0525@foxmail.com

² Riseye Intelligent Technology Co., Ltd., Shenzhen, China
qjkong@riseye.ai

Abstract. With the improvement of the precision of industrial cameras and the popularity of applications, the visual inspection model method of deep learning is more and more widely used in the field of industrial inspection. According to the analysis of the basic needs of the current industrial inspection field, we found that many need to carry out rapid detection and iteration of small data sets. We propose a platform model that is more in line with current industrial inspection production. And the model is based on many convolutional neural network architectures, including spatial transformer network, Faster-RCNN, YOLO, etc.

Keywords: Industrial detection · Deep learning · Rapid iteration

1 Introduction

In recent years, more and more deep learning visual inspection methods have been applied to industrial production. But in fact, there are many difficulties. The purpose of using deep learning method is to replace traditional manual testing to improve production efficiency. According to the different products to be detected, and the difference between imaging and lighting, generally, before the application, it is necessary to test the effect of the detection on the small data set. So the platform is supposed to quickly determine whether such products can be tested for deep learning detection model. The detection of traditional computer graphics accounts for a large proportion. However, it has the disadvantages of high lighting requirements and a single detection environment.

According to the real cases we have obtained, many times merchants only allow a small number of images to be tested, even single digits. So at least, we should prove that the model is useful and the evaluation should be finished as soon as possible.

In order to overcome the difficulties above, we combine the previous technical methods, select the appropriate ones, and propose our own platform. Spatial transform network [11] is considered to put the object in the right place, and several object detection methods and data augmentation algorithms are combined together.

Regarding the general detection with high-precision requirements, the industry uses traditional fixed detection methods to pursue accuracy rather than the versatility of the task, such as feature extraction and location with traditional visual algorithms. However, in many cases, lighting and cameras do not fully meet the requirements. Therefore, in the case where the detection algorithm is difficult to meet the standard, it can only be replaced by humans. Our platform can play a better role in this situation. For a single simple industrial product, only 100 to 200 images are required to accurately locate and detect such serious defects as scratches, creases, etc. Besides, due to the characteristics of convolutional neural networks, our method has a higher fault tolerance for the changes of detecting environment, e.g. the detection results will not be easily affected just by a slight change in brightness.

The contribution of this paper can be summarized as follows:

- (i) We first adopt the existing STN to our classification and object detection tasks in industrial field, which can be seen as a way of data augmentation and solve the problem of too little industrial data. Besides, this method has a fast iteration speed and has a good application prospect in industrial detection.
- (ii) We then propose our industrial defect detection platform which has a friendly user interface and a wealth of detection capabilities, including image classification and object detection tasks. Anyone can easily train the model and judge the pros and cons of the model by the built-in evaluation metrics so that the best model can be applied to the actual industrial detection task. The advantages of our platform are simple operations, fast iteration speed and robustness to the lighting environment.

2 Related Work

Generally, there are four major tasks about image recognition in computer vision, i.e. classification, localization, detection and segmentation. We only discuss image classification and object detection in this paper, which are both supervised learning tasks learning a function that maps an input to an output based on example input-output pairs. However, in many cases, the labeled data is scarce, and the unlabeled data is quite rich. The cost of manually labeling data is very large, so active learning has emerged. Active learning queries the most useful unlabeled samples through a certain algorithm, and then label them by experts. Next, the labeled samples are used to train the classifier to improve the accuracy of the model. Many active learning methods [18, 28, 30–32] have been proposed recently.

For image classification, the amount of the samples plays a critically important role in achieving higher performance. Thus, data augmentation, applying some transformation to the original dataset, is proposed and widely used to generate more synthetic samples. There has been a variety of data augmentation methods so far, e.g. random cropping, horizontal flipping [13] and some generative models [6, 26, 33]. In other words, it is used to avoid overfitting problem,

or called weak generalization ability, caused by too little data, which is also a focus in industrial detection. Besides, SPMTL [17] attempts to jointly learn the tasks by taking into consideration the complexities of both tasks and instances to improve the generalization ability of the model.

Object detection is another important topic in computer vision. It has a great development in recent years and can be summarized as two-stage and one-stage object detection methods. Two-stage object detection methods like R-CNN [5] is well known for its breakthrough progress in object detection. R-CNN was originally proposed to apply large convolutional neural networks to bottom-up proposal regions to locate and segment objects. This method first creates some pre-selected areas called bounding boxes, and then score the boxes to obtain the most possible position of the detected target. After a series of developments, Fast R-CNN [4] and Faster R-CNN [24] have been successively proposed. Various image detection methods based on CNN have emerged in an endless stream. Mask R-CNN [8] integrates image detection and segmentation, and also achieves good performance.

On the other hand, one-stage detection is a better choice if the real-time and feedback speed need to be taken into account. YOLO [21–23] and SSD [19] are the representatives. They can quickly detect and locate the target in the image through regression and feature processing, which is the main algorithm for video detection [29]. A further application of object detection is face recognition [16]. Person re-identification is one of the research directions, and [27] combined the structure of human and proposed a novel framework for addressing the problem of cross-view spatial misalignments in person Re-ID.

However, when applied to the industrial field, the methods mentioned above seem to be cumbersome. For VOC [3], CIFAR-10 [12], ImageNet [14] and other large datasets with high-quality images, there are enough number of samples to facilitate the training of models. If the project needs to land in the industrial field, the first problem needed to overcome is how to obtain acceptable detection results with small number of samples, and the tolerance of the manufacturers for the results. As a consequence, we combine the characteristics of the previous models and integrate them to overcome the problems faced by industrial detection to some extent, i.e., less training samples and quick iteration requirements. STN [11] is a tiny and well-designed network. It has been applied to natural environment text detection [2] in many studies before and is placed before the detection network, so as to make it lightweight and fast to detect tile and distortion targets. Here we use this idea to help overcome some of the difficulties in industrial detection, i.e. we can insert STN before the classification network or the detection network so that the target can be reshaped and be good for recognition. Finally, the results are displayed clearly by our platform.

3 Proposed Platform

The overall architecture of our platform can be illustrated in Fig. 1. The platform is divided into two major sections, namely image classification and object

detection. Each section includes at least three phases of training, testing and detection, while the object detection section has an additional phase to create data sets called image annotation. The platform runs on a Ubuntu system.

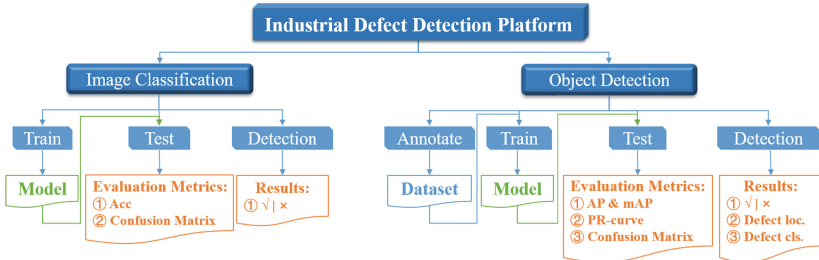


Fig. 1. The overall architecture of our industrial defect detection platform, where the green graphics denotes the model learnt from data set and the orange graphics denotes the output of testing and detection phase. (Color figure online)

3.1 Image Classification

Training. In training phase, we embed a spatial transformer network before a simple CNN to let the network learn how to transformer the input image so that the model can be better trained and the accuracy of classification can be further improved, which can be seen as a data augmentation method. We offer a variety of CNN backends to choose from, including the simple LeNet [15], AlexNet [13], VGG [25], and popular ResNet [9], DenseNet [10], ResNeXt, etc. We record the loss and accuracy values during training and draw a curve to dynamically monitor the progress.

Testing. In testing phase, we evaluate the trained model by two metrics, namely overall accuracy and confusion matrix. Regarding the confusion matrix, it is used to describe the performance of a classifier, including the precision and recall of each category. Each column of it represents the category to predict, with the total number of each column representing the number of samples predicted for that category. Similarly, each row of the matrix represents the ground truth category and the total number of each row represents the number of samples for that category. The number in each column indicates how many samples are predicted as this category.

Detection. In detection phase of image classification, we directly feed the image to the model to output the probability for each category and take the highest as the final result of the classification.

3.2 Object Detection

Annotation. Training data for object detection requires images and the corresponding ground truth annotation result. So this phase is designed to create datasets. The labeling results are stored as separated XML files according to the rules of VOC2007 [3].

Training. In training phase, we support several network architectures for object detection, including both one-stage and two-stage methods. Considering the rapid development of this field, we choose the architectures with relatively the highest efficiency and the best performance so far, namely Faster R-CNN [24] and Mask R-CNN [8] for two-stage detection and YOLOv3 [23] for one-stage detection. We monitor the progress of training by epoch-loss&accuracy curve mentioned in Sect. 3.1.

Testing. In testing phase of object detection, we choose AP (Average Precision), mAP (mean Average Precision), PR-curve (Precision-Recall curve) and confusion matrix as evaluation metrics. Note that AP and PR-curve are used to evaluate each category, while mAP and confusion matrix evaluate the overall performance of the model.

Detection. In detection phase, we feed the raw image into the trained model and the output is the annotated image with targets detected, including the bounding box indicating location and the category of each target.

The details will be displayed in the case study in Sect. 4.

4 Case Study

In this section, we present a case study on gear detection that demonstrates our platform's ability to solve such problems. Here we compare with traditional visual algorithms to highlight the advantages of our platform.

4.1 Traditional Visual Algorithm

We use OpenCV to process and detect defects, whose full name is open source computer vision library. As the name suggests, it mainly provides graphics processing and implementation methods for different applications. The core module that we mainly use to process images is the Imgproc module, which includes linear or nonlinear filtering operations, morphological operations, thresholding operations, etc. In addition, image matrix transformation operations, line detection, edge detection and other derivative detection methods are also the key point.

The image of the gear that needs to be processed is shown in Fig. 2(a). Since the gear we selected is centrally symmetric, the universality is not so high, but the advantage is that the features are clear. Symmetry defects or image

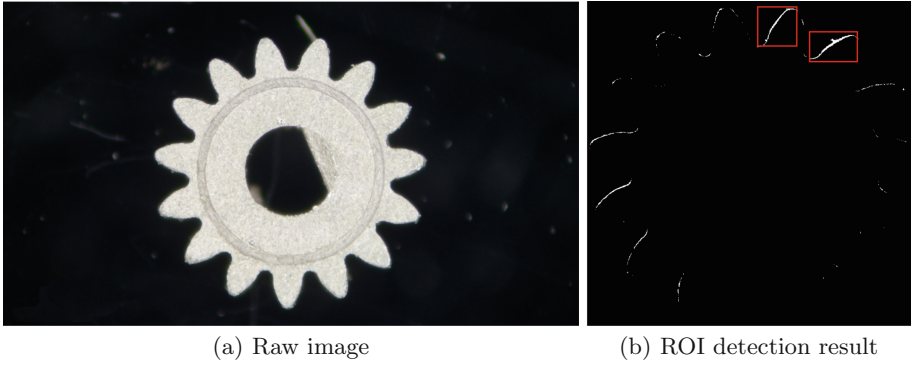


Fig. 2. Input raw image and output result image.

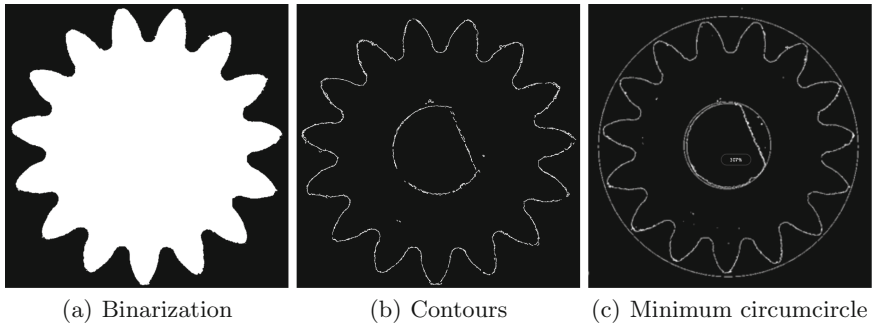


Fig. 3. Preprocess for the gear image.

changes caused by shooting can be quickly detected. Here we take the surface of the part as the research object. The main idea of our gear defect detection is that, considering the gear is a central symmetrical figure, we calculate the gear template and then calculate the difference between the template and the original image to locate the defect.

Observing the contrast of the image, the noises are not so obvious that we can directly select a fixed threshold [20] to binarize the image and the result can be seen in Fig. 3(a). We can find that the noises have been eliminated and the contours can then be extracted as Fig. 3(b). The inner contour is relatively stable compared to the rough outer contour, so we can select the center of circumcircle of the inner contour as the center of rotation. All we need to do is find the smallest circumcircle of the contours in the center of the gear in Fig. 3(b). We can use the method proposed by [7], which is quite common in the task of gesture recognition. The result is illustrated in Fig. 3(c). Now we get two main circles. We record the center of the inner circle as p_1 and the radius as r_1 . We record the center of the outer circle as p_2 and the radius as r_2 . Then we can extract ROI area from these data, namely $Rect(p_1.x - r_2, p_1.y - r_2, r_2 * x, r_2 * x)$. After

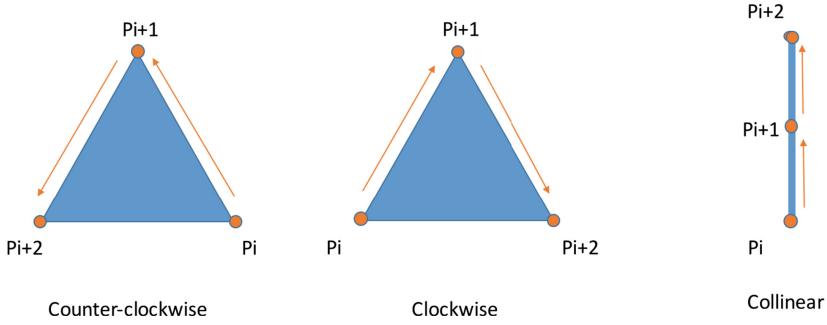


Fig. 4. The direction of the rotation

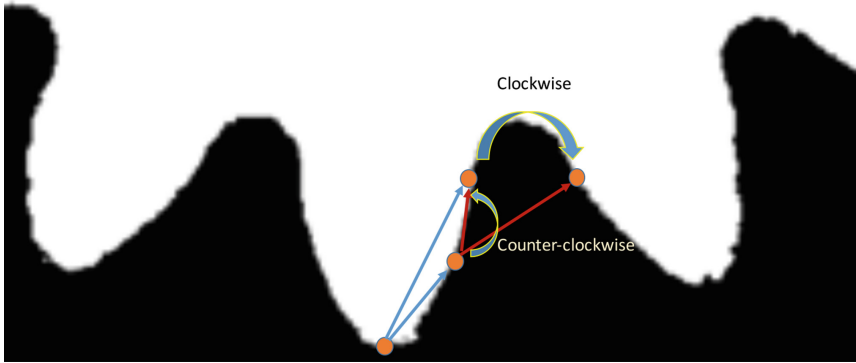


Fig. 5. The situation in a raw image.

the ROI area is extracted, the pixel points that need to be operated are greatly reduced, which is more convenient.

Next, we need to count the number of teeth of the gear. The premise is to determine the relative characteristics of the sawtooth relative to other shapes and to count according to the uniquely determinable feature. We can think of the gears that need to be counted as convex hull, and then use the method of convex hull detection [1] to judge the convex part of the gear. This method is convenient and clear. We find the lowest point in the image set to P_0 . Then, the other points and the cosine values of the vector and the X-axis are calculated. Based on this value, all the points on the permutation graph are arranged from large to small $P_1, P_2 \dots P_i$. After that we should judge the the direction of vector rotation from vector $P_i P_{i+1}$ to vector $P_{i+1} P_{i+2}$. If the rotation is counterclockwise, then this point is the constituent point of the convex hull. And in Fig. 4, we show the direction of the rotation. It can be judged by their cross product. $C.z = P_{i+1}.x * P_{i+2}.y - P_{i+1}.y * P_{i+2}.x$. And the positive and negative values of $C.z$ correspond to clockwise and counterclockwise, respectively. The situation is shown in Fig. 5.

Finally, we rotate the gear at some specific angles based on the number of teeth to find the average value, which we take as a template for the gear. We then calculate the difference between the template and the original raw image and binary the result to filter out the interference. We judge the area of the connected area beyond a certain threshold as a defect and mark it with the bounding box, and the result is shown in Fig. 2. Of course, this is only the result of the ROI, so we also need to mark the defects in the original raw image according to the coordinates of the bounding box obtained.

4.2 Deep Learning Algorithm

Now we present our deep learning method on solving this kind of problem. Note that this is an object detection task, taking into account the requirements of detection speed and accuracy, we choose to use YOLOv3 under the trade-off.

With our platform, what we need to do is to annotate the target for each gear image, namely marking the bounding boxes on the image and set the category for each box. Next, we only need to input the images and their corresponding annotation data into the network. After a few hours' training, computer has automatically extracted and learned the features through the deep neural network formed by the convolutional layer. Finally, we can get a model output by our platform that stores important information for solving gear defect detection tasks.

The model can be evaluated in the testing phase mentioned in Sect. 3.2. AP and overall performance of the model is shown in Fig. 6. As can be seen from the figure, there are a total of 100 test images and all of them are detected with a threshold of 0.8. The number of missed and false detection are both 0,

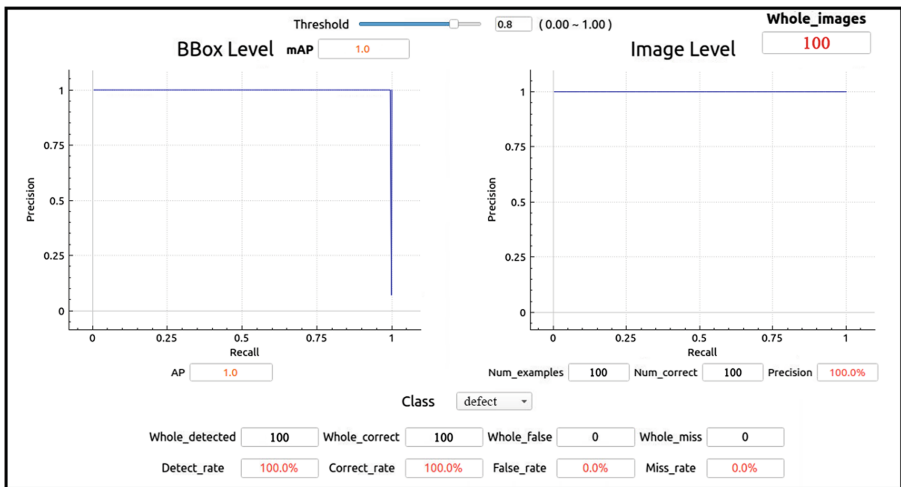


Fig. 6. Evaluation metrics by our platform.

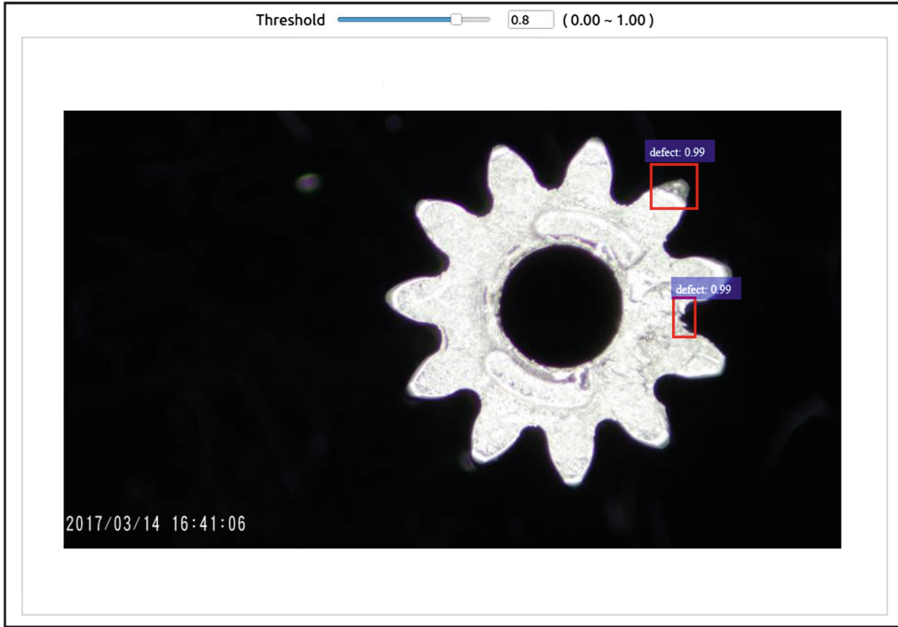


Fig. 7. An example of detection result by our platform.

and the detection accuracy reaches 100%. The data for this batch of gear defect detection has only one category, namely 'defect', whose AP (Average Precision) also reaches 100%, indicating that all the bounding boxes exist in the testing data are detected. The result is a good example of how our model (our platform) can handle this task.

An example of detection result is illustrated in Fig. 7. It can be found that when the threshold is set to 0.8, a total of two defects are detected, and the scores are both 0.99, which indicates that our model has high confidence in the judgment it makes.

5 Conclusion

Comparing the traditional method with the platform we proposed, it can be clearly seen that deep learning can effectively reduce the difficulty of industrial appearance defect detection and greatly improve the efficiency of algorithm development, because we do not have to spend a lot of time on researching and testing algorithms. In addition, the detection efficiency of our framework is also superior to traditional algorithms, reaching 20 fps. Another point is that our platform is more robust to different environmental conditions, such as not having a big impact on the results due to changes in the lighting configuration.

References

1. Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **22**(4), 469–483 (1996)
2. Bartz, C., Yang, H., Meinel, C.: STN-OCR: a single neural network for text detection and text recognition (2017)
3. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. *Int. J. Comput. Vision* **88**(2), 303–338 (2010)
4. Girshick, R.B.: Fast R-CNN. *CoRR* (2015)
5. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation (2013)
6. Goodfellow, I.J., et al.: Generative adversarial nets. In: *International Conference on Neural Information Processing Systems* (2014)
7. Har-Peled S, M.S.: Fast algorithms for computing the smallest k -enclosing circle (2005)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *CoRR* (2015)
10. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks. *CoRR* (2016)
11. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks (2015)
12. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *International Conference on Neural Information Processing Systems* (2012)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2009)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: *Proceedings of the IEEE* (1998)
16. Li, C., Liu, Q., Dong, W., Zhu, X., Liu, J., Lu, H.: Human age estimation based on locality and ordinal information. *IEEE Trans. Cybern.* **45**(11), 2522–2534 (2017)
17. Li, C., Fan, W., Yan, J., Dong, W., Liu, Q., Zha, H.: Self-paced multi-task learning (2016)
18. Li, C., Wang, X., Dong, W., Yan, J., Liu, Q., Zha, H.: Joint active learning with feature selection via cur matrix decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(99), 1382–1396 (2018)
19. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
20. Ohtsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (2007)
21. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: unified, real-time object detection (2015)
22. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242* (2016)
23. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. *arXiv* (2018)

24. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks (2015)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *Computer Science* (2014)
26. Yang, X., Deng, C., Zheng, F., Yan, J., Liu, W.: Deep spectral clustering using dual autoencoder network (2019)
27. Yang, S., Lin, W., Yan, J., Xu, M., Wu, J., Wang, J.: Person re-identification with correspondence structure learning (2015)
28. Zhang, X.Y., Wang, S., Yun, X.: Bidirectional active learning: a two-way exploration into unlabeled and labeled data set. *IEEE Trans. Neural Networks Learn. Syst.* **26**(12), 3034–3044 (2017)
29. Zhang, X.Y., Shi, H., Li, C., Zheng, K., Duan, L.: Learning transferable self-attentive representations for action recognition in untrimmed videos with weak supervision (2019)
30. Zhang, X.Y., Shi, H., Zhu, X., Li, P.: Active semi-supervised learning based on self-expressive correlation with generative adversarial networks. *Neurocomputing* **345** (2019). <https://doi.org/10.1016/j.neucom.2019.01.083>
31. Zhang, X.Y., Wang, S.P., Zhu, X.B., Wu, G.J., Wang, Y.P.: Update vs. upgrade: modeling with indeterminate multi-class active learning. *Neurocomputing* **162**, 163–170 (2015)
32. Zhang, X.Y.: Interactive patent classification based on multi-classifier fusion and active learning. *Neurocomputing* **127**, 200–205 (2014)
33. Zhu, X., Zhang, X., Zhang, X.Y., Xue, Z., Wang, L.: A novel framework for semantic segmentation with generative adversarial network. *J. Vis. Commun. Image Represent.* **58** (2018). <https://doi.org/10.1016/j.jvcir.2018.11.020>