



Cascading Convolutional Neural Network for Steel Surface Defect Detection

Chih-Yang Lin¹(✉), Cheng-Hsun Chen¹, Ching-Yuan Yang¹,
Fityanul Akhyar¹, Chao-Yung Hsu², and Hui-Fuang Ng³

¹ Department of Electrical Engineering, Yuan-Ze University, Taoyuan, Taiwan
andrewlin@saturn.yzu.edu.tw

² Automation and Instrumentation System Development Section,
China Steel Corporation, Kaohsiung, Taiwan

³ Department of Computer Science, FICT, Universiti Tunku Abdul Rahman,
Petaling Jaya, Malaysia

Abstract. Steel is the most important material in the world of engineering and construction. Modern steelmaking relies on computer vision technologies, like optical cameras to monitor the production and manufacturing processes, which helps companies improve product quality. In this paper, we propose a deep learning method to automatically detect defects on the steel surface. The architecture of our proposed system is separated into two parts. The first part uses a revised version of single shot multibox detector (SSD) model to learn possible defects. Then, deep residual network (ResNet) is used to classify three types of defects: Rust, Scar, and Sponge. The combination of these two models is investigated and discussed thoroughly in this paper. This work additionally employs a real industry dataset to confirm the feasibility of the proposed method and make sure it is applicable to real-world scenarios. The experimental results show that the proposed method can achieve higher precision and recall scores in steel surface defect detection.

Keywords: Fully convolutional networks · Defect detection · SSD · ResNet

1 Introduction

The billet is an upstream product of the rod and wire from Sinosteel. The process of producing billet from casting to production involves cooling, sand-blasting, rusting, inspection, grinding and heating, and finally rolling into strips. Billets are approximately 145 mm × 145 mm in size, and can be supplied to strip and wire factories for rolling into strip steel, wire rods and linear steel. However, inspection is necessary to ensure the quality of the product before it is sent out.

The surface temperature of billets reaches as high as 700 to 900° [1] in the production environment. These conditions make defect detection on billets difficult to achieve. Traditional billet defect detection methods are divided into visual inspection [2, 3] and magnetic particle inspection [4]. However, visual inspection is more cost and time efficient; therefore, we will only focus on visual inspection in this paper. The types

of defects found can indicate the cause of defect formation and be used to improve the steelmaking process, since different defects have different causes.

In this paper, we develop a billet defect detection technology based on convolutional neural network. We propose a hierarchical structure to detect defects with revised SSD and ResNet50 [5, 6]. The experimental results show the effectiveness of the proposed method.

2 Architecture Overview

2.1 Structure of SSD

With the rise of convolutional neural networks, many models have evolved, such as Faster RCNN [7], Mask RCNN [8], Single Shot Multibox Detector (SSD) [9], and You Only Look Once (YOLO) [10]. All of these models have object detection capabilities. Among them, we chose the SSD300 version as the basic model. The reasons that we selected SSD300 are as follow:

- Faster RCNN and Mask RCNN are two-stage methods, which means that the training process is performed in two steps. In contrast, SSD and YOLO are one-stage methods, which are more efficient.
- The detection speed is better than that of other models. According to the author's paper, the detection speed of SSD300 is 59 FPS (frames per second).
- The architecture of SSD300 is simpler than other models' architectures and easier to adjust.
- SSD has multi-scale predictions.

The original SSD300 contains anchor boxes that are a combination of horizontal and vertical rectangles as shown in Fig. 1(a). In this work, we only use three horizontal rectangles as shown in Fig. 1(b). When there are many prediction boxes on an object, as shown in Fig. 2(a), non-maximum suppression (NMS) in SSD can solve this problem as shown in Fig. 2(b).

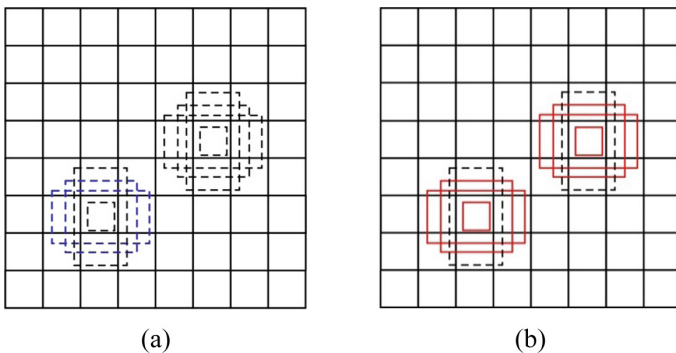


Fig. 1. (a) Original anchor box. (b) Customized anchor box.

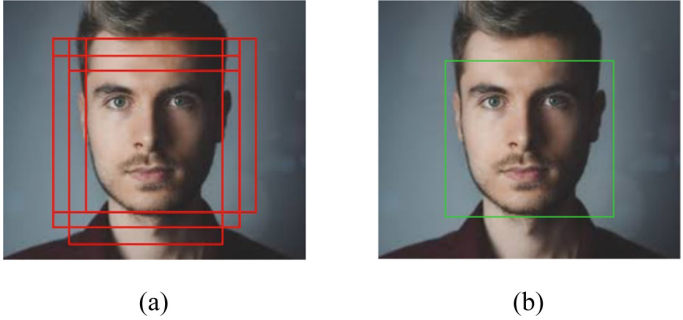


Fig. 2. (a) Bounding boxes before NMS. (b) Bounding box after NMS.

In Figs. 3–5, we present our revisions to the SSD architecture based on characteristics of collected defect images. Billet defects are mainly small ones. In Fig. 4, a 75×75 feature map is added to convolutional block 3 of the VGG16 layer, and the last two feature maps (3×3 and 1×1) are removed. In order to compare advantages and disadvantages of various SSD structures, the original SSD module (SSD300) and modified SSD module (revised-SSD300) will be trained. In addition, the revised-SSD300 will be extended to a revised-SSD600 with an input size of 600×600 as shown in Fig. 5. Therefore, in total, three models will be trained for comparison.

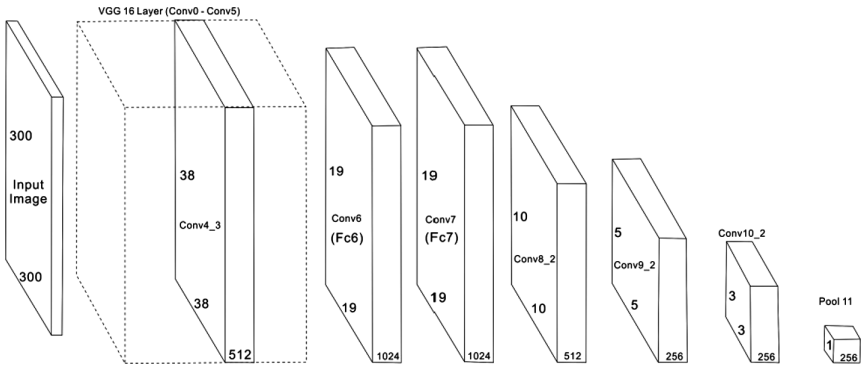


Fig. 3. SSD300 architecture.

2.2 Introduction of SENet and ResNet

In our main task, we need to detect two defects, called “sponge” and “scar” defects. The task of SSD is to determine whether the defects exist and where they are. We also added Squeeze-and-Excitation Net (SENet) [11] structure in our model to boost the

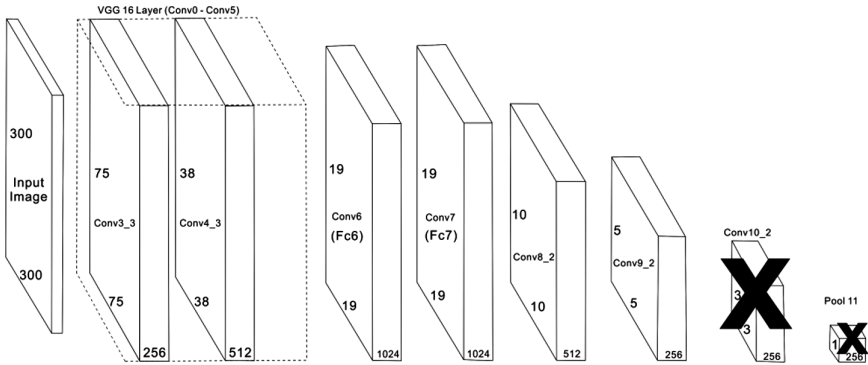


Fig. 4. Revised-SSD300 architecture.

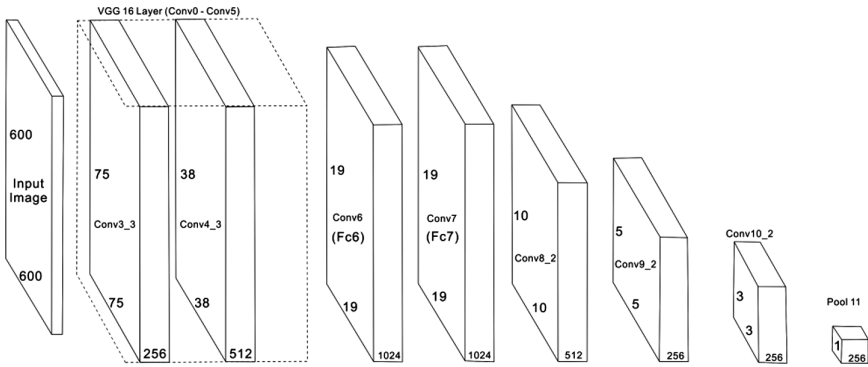


Fig. 5. Revised-SSDSE600 architecture.

results with adaptive weights for each feature map. SENet is not a complete network structure, but rather a small architecture in between convolution blocks. When SENet is applied, our method is called Revised-SSDSE, which is shown in Fig. 5.

Sometimes, another non-defect factor, called “rusty factor” as shown in Fig. 9(c), will be present in the dataset. The rusty factors, which are not defects, have various shapes and features and significantly affect our results. In order to detect rusty factors in the dataset, the 3×3 and 1×1 layers must be added back to the revised SSD network.

After determining the existence and location of defects, ResNet should identify the name of the defect. In this paper, we use ResNet50 [12] and classify three categories of defects as shown in Fig. 9. The defect from SSD will be resized to 224×224 to fit the input size for ResNet50. The combination of revised-SSDSE600 and ResNet50 forms the complete hierarchical structure as shown in Fig. 6.

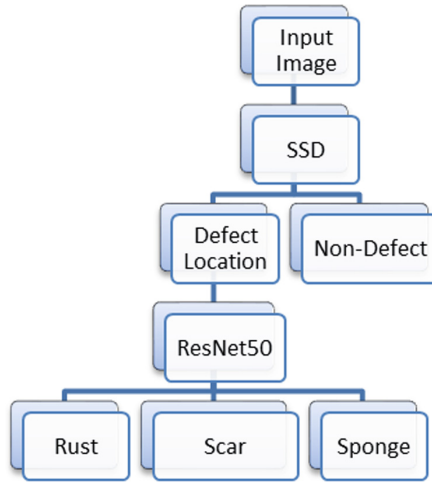


Fig. 6. Final hierarchical structure.

3 System Requirements

3.1 Hardware and Software

The hardware and software environment used in this paper is given in Table 1. The software part of the system includes Anaconda and GPU environment settings.

Table 1. Hardware and software environment.

Environment	Description
Operating System	Windows 10
Central Processing Unit (CPU)	Intel(R) Core(TM) i7-7700 CPU @ 3.60 GHz
Graphics processing unit (GPU)	NVIDIA GeForce GTX 1080 Ti 12G
CPU Memory	8 GB
Distribution	Anaconda v4.1.0
Programming Language	Python v3.5.4
Python Library	TensorFlow v1.8.0, Keras v2.1.6, Matplotlib v2.2.0, PIL v5.1.0
Editor	Visual Studio Community 2017
GPU Platform	NVIDIA CUDA 9.0
GPU Library	NVIDIA cuDNN 7.0

3.2 Data Annotation

We use LabelImg v1.6.0¹ tool to mark defect locations and non-defect classes in the dataset for the SSD model. LabelImg supports several operating system platforms, like Windows, Linux and Mac OS X. In this work, we use a Windows environment. After the labeling process is completed, the label result is saved in an XML format.

4 Experimental Results

The detection results are affected by camera types, illumination, number of defective samples, and other factors. The training process is performed as follows.

- Collect various defect samples.
- Mark the defect samples and generate corresponding XML files containing defect information.
- Train marked defect samples through the neural network structure and save the training results.

Table 2. Initial results of models.

Model	Sponge	ScarRemain
SSD 300	60%	85%
revised-SSD300	74%	95%
revised-SSD600	76%	95%

4.1 Initial Test

In the initial test, we prepared defect data with 464 Scar and 246 Sponge images in the dataset, 10% of which were validation and 90% of which were training data. The experimental results are shown in Table 2. The results in Fig. 7 show that the performance of the revised-SSD300 is similar to that of the revised-SSD600, but better than SSD300. There are too many redundant boxes when SSD300 is applied, as presented in Fig. 7b.

We test the daily images provided by the onsite database and used the following parameters as an accurate benchmark for calculating the system performance [13]:

- True Positive (TP).
- True Negative (TN).
- False Positive (FP).
- False Negative (FN).
- Precision (P) in Eq. (1).
- Recall (R) in Eq. (2).

¹ <https://github.com/tzutalin/labelImg>.

- F-Measure is a comprehensive evaluation index, which is used to understand whether two values of Precision and Recall are good, as shown in Eq. (3).

$$P = TP / (TP + FP). \tag{1}$$

$$R = TP / (TP + FN). \tag{2}$$

$$F\text{-Measure} = (2 \times P \times R) / (P + R). \tag{3}$$

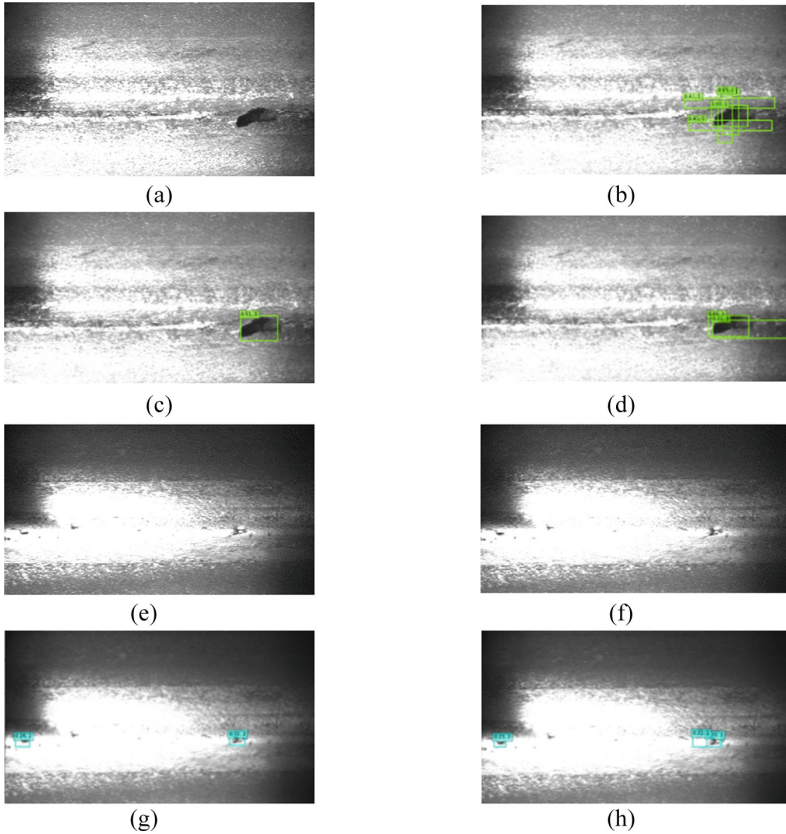


Fig. 7. (a) Original image. (b) SSD300 prediction results. (c) Revised-SSD300 prediction results. (d) SSD600 prediction results. (e) Original image. (f) Original image. (g) Revised-SSD300 prediction results. (h) SSD600 prediction results.

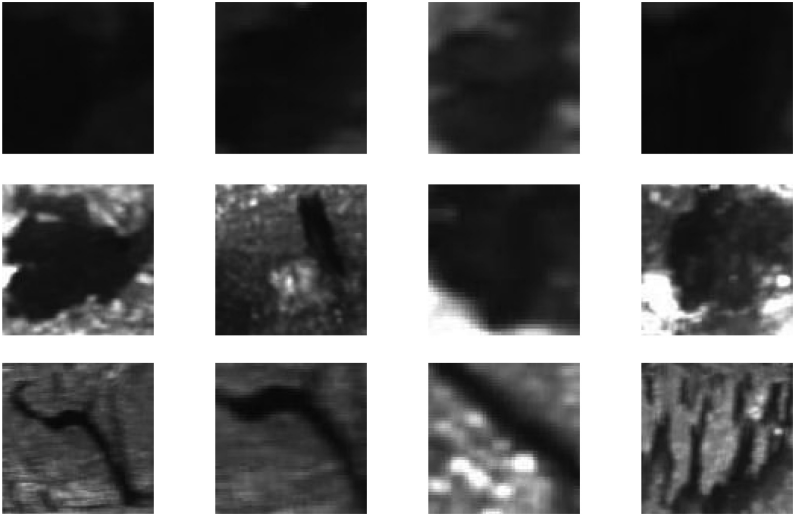


Fig. 8. Defect samples produced from SSD directly.

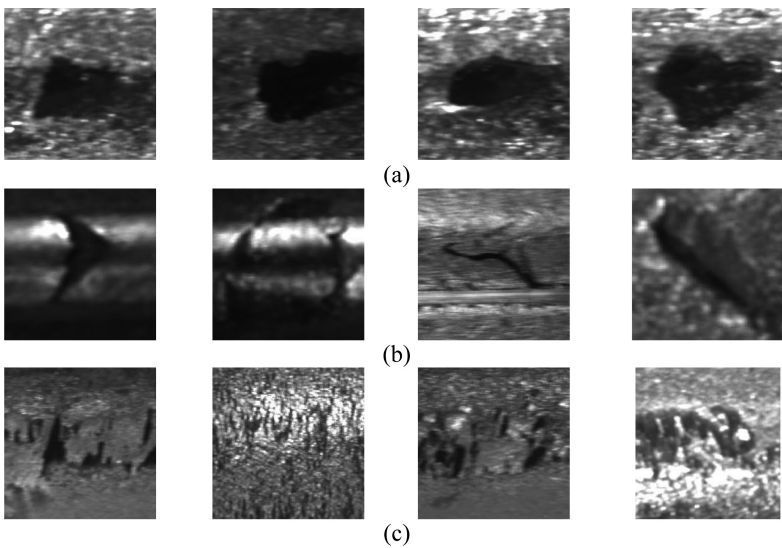


Fig. 9. Extension of defect range. (a) Scar. (b) Sponge. (c) Rust.

4.2 Final Test

According to Tables 3 and 4, after a seven-day training period, the highest precision and recall of the revised-SSD300 were 100% and 77.6%, respectively. The revised-SSD600 had a better recall due to its high-resolution images. However, the

combination of the revised-SSDSE600 and ResNet50 achieved the highest precision and recall rates.

Table 3. Final test results of revised-SSD300.

Time (day)	TP	TN	FP	FN	P (%)	R (%)	F-Measure (%)
1	302	19	1	87	99.6	77.6	87.2
2	142	13	4	95	97.2	59.9	74.1
3	102	38	19	88	84.2	53.6	65.5
4	97	38	11	75	89.8	56.3	69.2
5	35	3	0	37	100	48.6	65.4
6	54	5	0	43	100	5.56	71.5
7	73	5	1	51	98.6	58.8	73.7

Table 4. Final test results of revised-SSD600.

Time (day)	TP	TN	FP	FN	P (%)	R (%)	F-Measure (%)
1	315	19	1	74	99.6	80.9	89.3
2	183	13	4	54	97.8	77.2	86.3
3	139	38	19	51	87.9	73.1	79.8
4	105	38	11	67	90.5	61.0	72.9
5	45	3	0	27	100	62.5	76.9
6	61	5	0	36	100	62.8	77.2
7	89	5	1	35	98.8	71.7	83.1

Table 5. Final test results of the combination of revised-SSDSE600 and ResNet50.

Time (day)	TP	TN	FP	FN	P (%)	R (%)	F-Measure (%)
1	353	19	1	36	99.7	90.7	95.0
2	226	13	4	11	98.2	95.3	96.7
3	179	38	19	11	90.4	94.2	92.2
4	138	38	11	34	92.6	80.2	85.9
5	66	3	0	6	100	91.6	95.6
6	95	5	0	2	100	97.9	98.9
7	118	5	1	6	99.1	95.1	97.1

Note that Fig. 10 shows that if we used defect bounding boxes directly from SSD for ResNet as shown in Fig. 8, the training process was hard to converge because the bounding boxes were too fitted to the defects. Therefore, we enlarged the range of the bounding boxes as shown in Fig. 9. After extending the bounding box, the training process could converge, which enabled the performance in Table 5 to be achieved.

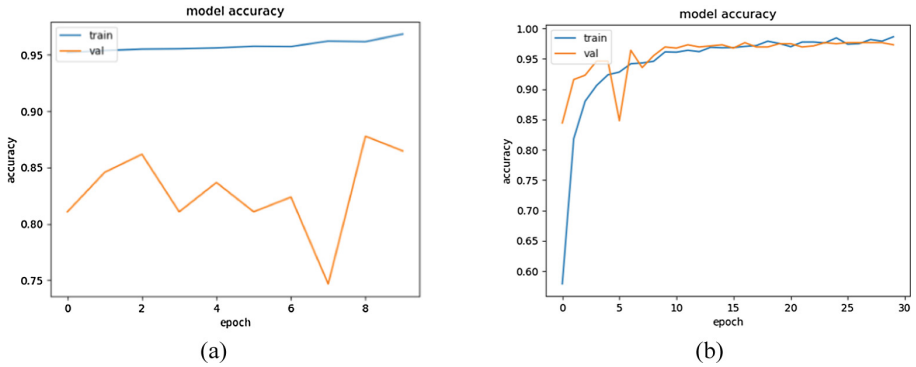


Fig. 10. Train results of revised-SSD600 with ResNet50. (a) With first samples of defect. (b) With second samples of defect.

5 Conclusions

In this paper, we design a hierarchical model to build a defect detection system for steel billets. We have modified the architecture of SSD by changing the sizes of feature maps and the sizes of anchor boxes to fit the shape of defects. The experimental results demonstrate the effectiveness of the proposed method. In further work, we will collect more rust defect images, because rusty types include many variations.

Acknowledgement. This work was supported by the Ministry of Science and Technology, Taiwan, under Grants MOST 106-2218-E-468-001, MOST 107-2221-E-155-048-MY3, and MOST 108-2634-F-008-001, and under Grants from China Steel Corporation RE106728 and RE107705.

References

1. Peacock, G.R.: Review of noncontact process temperature measurements in steel manufacturing. In: SPIE Conference on Thermosense XXI, pp. 171–189. SPIE, Florida (1999)
2. Yun, J.P., Choi, S., Kim, J.-W., Kim, S.W.: Automatic detection of cracks in raw steel block using Gabor filter optimized by univariate dynamic encoding algorithm for searches (uDEAS). *NDT E Int.* **42**, 389–397 (2009)
3. Duan, X., Duan, F., Han, F.: Study on surface defect vision detection system for steel plate based on virtual instrument technology. In: 2011 International Conference on Control, Automation and Systems Engineering (CASE), pp. 1–4. IEEE, Singapore (2011)
4. Balchin, N.C., Blunt, J.: *Health and Safety in Welding and Allied Processes*, 5th edn. Woodhead Publishing, Cambridge (2002)
5. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pp. 1097–1105. Curran Associates, Inc., Nevada (2012)

7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pp. 91–99. Curran Associates, Inc., Canada (2015)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. IEEE, Venice (2017)
9. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, vol. 9905. Springer, Cham (2016)
10. Redmon, J., Kumar Divvala, B., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. pp. 779–788. IEEE (2016)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141. IEEE, Salt Lake City (2018)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE (2016)
13. Powers, D.M.W.: Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. 2017 (2007)