

Anomaly Detection for Imbalanced Data via Deep Neural Network with Concentrated Loss

Yanli Wang^a, Xiaodong Wang^{a,1}, Xianwei Xu^a, Fei Yan^a, and Zhiqiang Zeng^a

^a College of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China

Abstract. Deep neural networks have recently been used to address surface anomaly detection in industrial quality control and have achieved much success. However, addressing the data imbalance problem, especially the Easy/Hard Examples (EHE) imbalance problem, remain a challenging task in anomaly detection. To alleviate this problem, we propose a two-stage convolutional neural network with a novel loss function, *i.e.*, concentrated loss function. Specifically, the concentrated loss function enables the model to pay more attention to hard examples and improve the quality of segmentation for imbalanced data. To verify the effect of our method, we implement our method on the surface anomaly detection dataset, *i.e.*, the KolektorSDD2 dataset. The experimental results show the superiority of our method over the other state-of-the-art approaches.

Keywords. Data imbalance, Anomaly detection, Deep learning, Quality control.

1. Introduction

As a crucial part of the process of industrial production, product quality inspection has attracted increasing attention. The product quality inspection directly affects production efficiency. To improve product quality and production efficiency, numerous surface anomaly detection methods have been proposed. Traditionally, machine vision-based techniques have been applied to product quality inspection processes [1, 2]. These methods utilize hand-crafted features to detect defects generally, difficult to cope with various types of defects.

With the development of big data and the enhancement of computing power, deep learning strategies have been applied to anomaly detection [3, 4, 5]. Deep learning methods can handle complex defect features well and adapt to changing scenarios in industrial applications. Concretely, anomaly detection networks can be divided into one-stage and two-stage frameworks. The one-stage network combines the segmentation and classification modules into a joint structure [6, 7].

However, this network relies on a large number of training samples, showing poor performance when training samples are lacking. On the contrary, the two-stage network is composed of two sub-networks, *i.e.*, the

¹ Corresponding Author. E-mail: xdwangjsj@xmut.edu.cn

segmentation network and the decision network. The segmentation network is used to locate the defect location. After the segmentation network, the decision network determines whether the defects exist based on the segmentation result. The two-stage networks can achieve excellent performance for addressing surface anomaly detection when there are fewer training samples. However, the data imbalance problem, which is frequently encountered in anomaly detection, restricts the two-stage network, and affects its detection accuracy.

For the real-world dataset, the data imbalance problem is almost inevitable [8]. This problem leads to the major examples overwhelm training and degenerate models [9]. Concretely, data imbalance in the anomaly detection domain can be divided into two categories, Class Imbalance (CI), and Easy/Hard Examples (EHE) imbalance [10]. CI represents the positive (defective) examples occupy a smaller proportion than negative (non-defective) examples on datasets. At present, there are many solutions to the problem of CI, and the specific methods are discussed in Section 2. Obtaining less attention, EHE imbalance refers to the number of hard examples that is much smaller than the number of easy examples. An easy example is shown in Figure 1 (a), which has an obvious defect target (marked by the red box) and is prone to be detected. In contrast, Figure 1 (c), shows a hard example that the defect target (marked by the red box) is difficult to be detected by anomaly detection algorithms, even for the human being. The detection of the hard examples on the dataset is the top priority to improve the accuracy of anomaly detection.

In this paper, we propose a concentrated loss (CL) to alleviate the EHE imbalance problem. CL makes the anomaly segmentation model pay more attention to the hard examples and shows excellent performance to its anomaly detection. We implement our method in a two-stage anomaly detection network and verify it with KolektorSDD2 (KSDD2) dataset². The experiments results, in Section 4.3, demonstrate the effectiveness of our method.

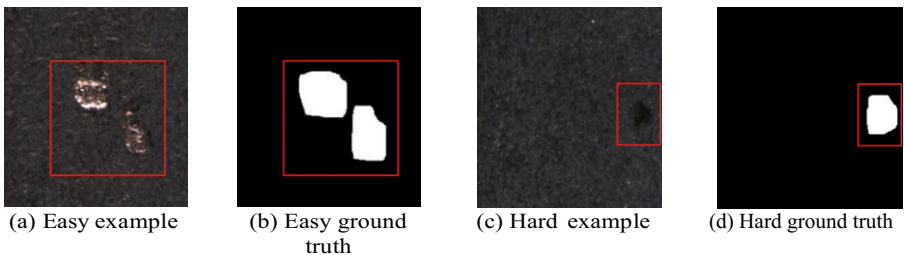


Figure 1. Illustration of the EHE imbalance problem. Subfigures (a) and (c) show the instance of easy example and hard example, respectively. Subfigures (b) and (d) show the corresponding defective region.

² <https://www.vicos.si/resources/kolektorsdd2>

2. Related work

Several related works explored the usage of deep learning methods in manufacturing industry anomaly detection [6, 7, 11, 12]. The work of Masci *et al.* [3] showed that deep learning methods are superior to traditional machine vision-based methods. They used a shallow network to detect steel anomalies and achieved excellent results. Further, Weimer *et al.* [4] evaluated several deep learning structures with different layers and conducted a more comprehensive study on deep network architecture. They are inefficient in classifying each individual image patch. A more efficient two-stage network for anomaly detection was proposed by Racki *et al.* [13]. The application of the two-stage network in anomaly detection tasks greatly improves detection accuracy [11]. However, the classification results of this method fluctuate greatly due to the data imbalance [10].

To address these problems, some methods have made a great contribution [14, 9, 15]. The earliest work by Chawla *et al.* [16] used over-sampling techniques to address the class imbalance. However, the algorithm has a certain degree of blindness when selecting neighbors. Further, Han *et al.* [17] improved based on the SMOTE algorithm [16] for refining the class imbalance of data. Furthermore, for the EHE imbalance problem of datasets, Shrivastava *et al.* [18] proposed online hard example mining (OHEM). The method retains examples with higher loss and ignores easy examples. Similarly, Lin *et al.* [9] proposed a novel loss function based on a one-stage network, *i.e.*, focal loss function. This loss function reduced the weight of a large number of easy examples in training and achieved excellent results for addressing the problem of data imbalance. Inspired by the success of the focal loss function on semantic segmentation, we propose the concentrated loss function for anomaly detection. The proposed strategy can improve the identification of hard examples on the two-stage network.

3. Proposed method

In this section, we show the details of our method. In practice, the EHE imbalance weakens the deep learning-based anomaly detection method. To reduce the impact of the EHE imbalance, we propose a concentrated loss (CL) function. CL down-weights easy examples and thus induces the model to focus on the hard examples. We implement CL on the segmentation network of the two-stage network. Through the combination of CL and the segmentation network, our model obtains better extraction performance on hard examples. As shown in Figure 2, the proposed network contains two sub-networks. The segmentation network locates the defects by extracting its mask, while the decision network based on the segmentation result determines whether there are defects on the surface of products. With two-stage cascade and biased minibatch sampling mechanisms, our two-stage network shows robust to class imbalance [9].

3.1. Network architecture

In this paper, we propose a novel two-stage network based on Tabernik *et al.* [11]. As shown in Figure 2, our network is consisting of two components, *i.e.*, the segmentation network and the decision network.

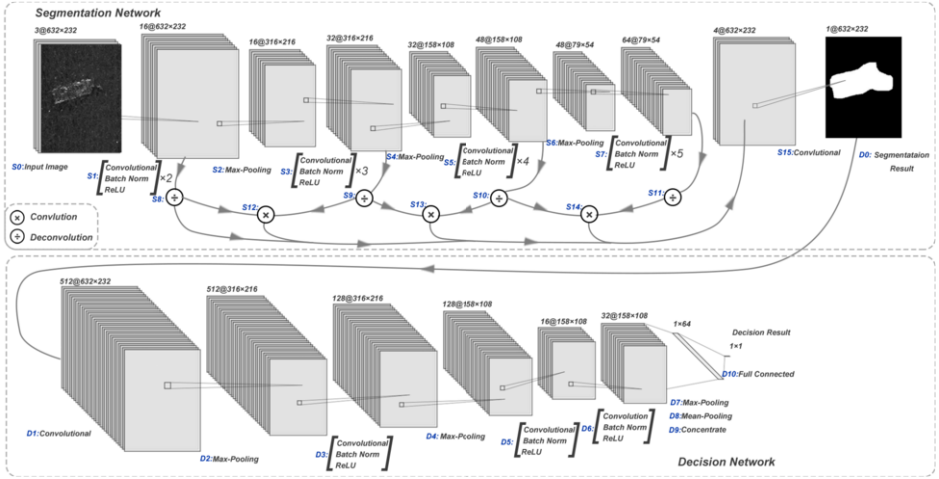


Figure 2. The structure of our neural network. In the figure we utilize “S number and D number” to represent layers of segmentation network and decision network.

For the segmentation network, we add convolutional layers (layer S7) based on the segmentation subnetwork proposed by Tabernik *et al.*, to enhance the feature extraction ability. However, with the increase of convolutional layers, the extracted defect information is more likely to be lost in forward propagation [19]. To save the extracted information, deconvolutional layers are adopted. Four deconvolutional layers (layers S8-S11) are inserted before each max-pooling layer and after the last convolutional layer. For convenience, we call the feature map after deconvolutional layers *e*-feature. These *e*-features are integrated by three convolutional layers *i.e.*, the layers S12-S14. Finally, we obtain the defect segmentation result by fusing the four integrated *e*-features via a convolution layer (layer S15). For the decision network, the channel of the input feature is adjusted to a single channel. It takes the segmentation result as input and outputs the predicted class of input image. The architecture details of our network are shown in Table 1.

3.2. Concentrated loss

Inspired by focal loss [9], we propose the concentrated loss (CL). CL is the product of basic loss function and concentrated factor as follow:

$$\mathcal{C}\mathcal{L}(\mathbf{Y}_n, \mathbf{P}_n) = (1 - \mathcal{I}(\mathbf{Y}_n, \mathbf{P}_n))^\gamma \mathcal{B}(\mathbf{Y}_n, \mathbf{P}_n). \quad (1)$$

Table 1. Architecture details for segmentation and decision sub-networks.

| Segmentation network | | | | Decision network | | | |
|----------------------|-----------------|--------|---------|------------------|---------------------|--------|---------|
| No. | Layer | Kernel | Channel | No. | Layer | Kernel | Channel |
| S0 | Input image | - | 3 | D0 | Segmentation Result | - | 1 |
| S1 | 2×Convolutional | 3×3 | 16 | D1 | Convolutional | 1×1 | 512 |
| S2 | Max-Pooling | 2×2 | 16 | D2 | Max-Pooling | 2×2 | 512 |
| S3 | 3×Convolutional | 3×3 | 32 | D3 | Convolutional | 5×5 | 128 |
| S4 | Max-Pooling | 2×2 | 32 | D4 | Max-Pooling | 2×2 | 128 |
| S5 | 4×Convolutional | 3×3 | 48 | D5 | Convolutional | 5×5 | 16 |
| S6 | Max-Pooling | 2×2 | 48 | D6 | Convolutional | 5×5 | 32 |
| S7 | 5×Convolutional | 3×3 | 64 | D7 | Max-Pooling | Global | 32 |
| S8 | Deconvolutional | 1×1 | 1 | D8 | Mean-Pooling | Global | 32 |
| S9 | Deconvolutional | 2×2 | 1 | D9 | Concatenate | - | 64 |
| S10 | Deconvolutional | 4×4 | 1 | D10 | Full connect | - | 54 |
| S11 | Deconvolutional | 8×8 | 1 | | | | |
| S12 - S14 | Convolutional | 3×3 | 1 | | | | |
| S15 | Convolutional | 3×3 | 1 | | | | |

In Eq. (1), $\mathbf{Y}_n \in \{0, 1\}$ denotes the ground truth mask of input samples, while $\mathbf{P}_n \in [0, 1]$ is denoted as the corresponding prediction of the model. Basic loss function $\mathcal{B}(\mathbf{Y}_n, \mathbf{P}_n)$ is a conventional loss function, which can be a single loss function like the Cross Entropy loss function or a combination of different loss functions. Concentrated factor $\mathcal{C}(\mathbf{Y}_n, \mathbf{P}_n) = (1 - \mathcal{I}(\mathbf{Y}_n, \mathbf{P}_n))^\gamma$ is a coefficient that dynamically changes according to the segmentation results. $\mathcal{I}(\mathbf{Y}_n, \mathbf{P}_n)$ is the intersection over union (IOU) of \mathbf{Y}_n and \mathbf{P}_n , while $\gamma \in [0, +\infty)$ is a coefficient, control the down-weighting degree for easy examples. IOU is the ratio of intersection and union of prediction and ground truth, which is formulated as

$$\mathcal{I}(\mathbf{Y}_n, \mathbf{P}_n) = \frac{2\mathbf{P}_n \mathbf{Y}_n}{(\mathbf{P}_n + \mathbf{Y}_n) - \mathbf{P}_n \mathbf{Y}_n + \epsilon}. \quad (2)$$

In the Eq. (2), ϵ is a small nonzero constant to prevent the denominator from being zero. The CL is visualized for several values of $\gamma \in [0, 5]$ in Figure 3. Take an instance of a hard example as input. When the network confronts a hard example, the segmentation network is difficult to extract defects, meaning the $\mathcal{I}(\mathbf{Y}_n, \mathbf{P}_n)$ is a small

value. In this case, the concentrated factor is close to 1 and CL is almost equal to the basic loss. In contrast, if the input image is an easy example, the segmentation network can segment the defect well with the $\mathcal{I}(\mathbf{Y}_n, \mathbf{P}_n)$ close to 1. In this situation, the concentrated factor is close to 0 and CL is much smaller than the basic loss. By adjusting the proportion of loss from the examples with different categories, the model can learn knowledge with balancing hard samples and easy samples.

In the experiment, we follow the previous study [20] adopting the weighted sum of Dice loss $\mathcal{D}(\mathbf{Y}_n, \mathbf{P}_n)$ and Cross Entropy loss $\mathcal{CE}(\mathbf{Y}_n, \mathbf{P}_n)$ to alleviate the class imbalance problem. Our loss function is formulated as follows:

$$\mathcal{CL}(\mathbf{Y}_n, \mathbf{P}_n) = (1 - \mathcal{I}(\mathbf{Y}_n, \mathbf{P}_n))^\gamma (\lambda \mathcal{D}(\mathbf{Y}_n, \mathbf{P}_n) + (1 - \lambda) \mathcal{CE}(\mathbf{Y}_n, \mathbf{P}_n)). \quad (3)$$

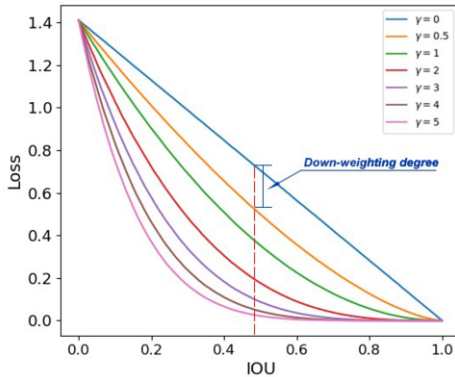


Figure 3. The visualization of CL. As shown in the figure, with the γ increases, the down weighting degree for easy examples increases as well.

4. Experiment

4.1. Dataset

In order to verify the effectiveness of our proposed algorithm, we utilize the KolektorSDD2 dataset as training and testing data. The dataset is constructed from images of defective production items that were provided and annotated by the Kolektor Group [12]. The images were captured in a controlled industrial environment and included different defect types such as scratches, small spots, and surface defects *etc.* This dataset contains 356 images with visible defects and 2979 images without any defect, in which each picture with a resolution of 632×232 pixels. Thereinto, the training set has 246 positive and 2085 negative images, and the test set has 110 positive and 894 negative images.

4.2. Performance metrics and implement detail

In the experiments, we mostly measure the performance in terms of the average precision (AP), which is calculated as the area under the precision-recall curve. AP is one of the commonly used metrics to measure the performance of classification models. A larger AP value indicates better classification capability.

The proposed method is implemented on Ubuntu 18.04 with Intel(R) Xeon(R) E5-2630 v4 CPU and NVIDIA GeForce GTX 2080 Ti GPU. The parameter γ in our method controls the down-weighting degree for easy examples, while the parameter λ adjusts the proportion of Dice loss and Cross Entropy loss. In this paper, we set γ and λ to 0.5 and 0.95, respectively. For both the segmentation and the decision sub-networks, we apply the random initialization of the standard normal distribution to the model parameters and use the Adam algorithm [21] to optimize the parameters. We train the segmentation model 200 epochs on the KSDD2 dataset with a learning rate of 0.1 and batch size of 2, while the decision model is optimized for 100 epochs.

4.3. Anomaly detection results

In this section, we verify that our method has the ability to alleviate the impact of the EHE imbalance. We propose the concentrated loss (CL) function to induce anomaly detection models to pay more attention to hard examples. The intuitive details are shown in Figure 4. It can be seen from Figure 4 that our method is accurate for hard and easy examples extraction.

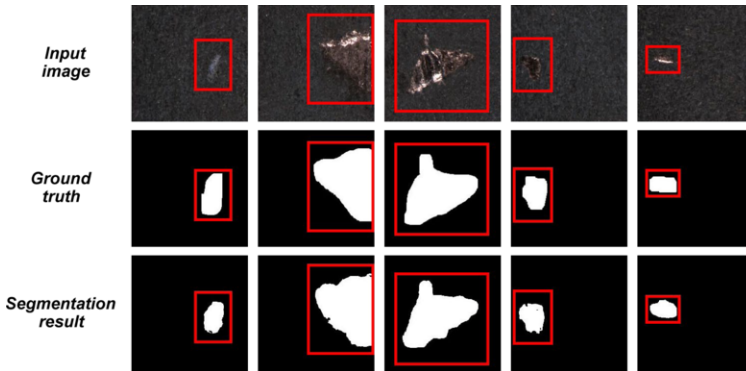


Figure 4. Segmentation results of our methods. The red boxes are adopted to label the location of defectregions.

Furthermore, we compare with two state-of-the-art methods, *i.e.*, Mixed supervision [12] and PSIC-Net [22]. Mixed supervision is a representation of the one-stage anomaly detection method, while PSIC-Net represents the two-stage anomaly detection method.

As shown in the rightmost column of Figure 5, our method obtains the AP of 96.1%, which is better than PSIC-Net and Mixed supervision 2.8 and 0.7 percentage points, respectively. In order to illustrate the effectiveness of CL, we conducted an ablation experiment on CL. Concretely, we only keep the basic loss by removing the Dice loss in Eq. (3), *i.e.*, $\lambda = 0$, for the segmentation network. The result is shown in the third column of Figure 5. Compared with $\lambda = 0$, the AP is improved by 0.4 percentage points when $\lambda = 0.5$. The above results prove the validity of our method.

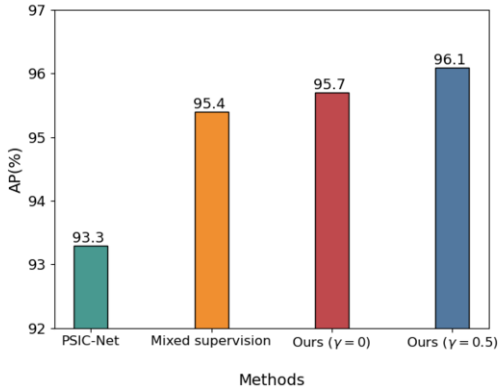


Figure 5. Anomaly detection comparison of related methods.

5. Conclusion

Product quality inspection is becoming more and more important for the manufacturing industry. In this article, we aim to alleviate the Easy/Hard Examples imbalance problem in product anomaly detection. Concretely, we propose the concentrated loss function to down-weight the loss of easy examples on the segmentation network and thus induce the anomaly detection model to pay more attention to hard examples. After deploying it with the two-stage neural network, we achieve excellent detection accuracy for both easy examples and hard examples. The comparison results show that our method achieves superior performance over other state-of-the-art.

Acknowledgement

This paper was supported by National Natural Science Foundation of China (Grant Nos.61871464, U1805264), National Natural Science Foundation of Fujian Province (Grant Nos.2020J01266, 2021J011186), the “Climbing”

Program of XMUT (Grant No.XPDKT20031), Scientific Research Fund of Fujian Provincial Education Department (Grant No. JAT200486), Program of XMUT for high-Level talents introduction plan (Grant No.YKJ19003R).

References

- [1] Y. Hu, C.-x. Zhao, A Novel LBP Based Methods for Pavement Crack Detection, *Journal of Pattern Recognition Research* 5 (1) (2010) 140–147. doi:10.13176/11.167.
- [2] W. Xu, Z. Tang, J. Zhou, J. Ding, Pavement crack detection based on saliency and statistical features, 2013 IEEE International Conference on Image Processing, IICIP 2013 Proceedings (September 2013) (2013) 4093–4097. doi:10.1109/ICIP.2013.6738843.
- [3] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, G. Fricout, Steel defect classification with Max-Pooling Convolutional Neural Networks, *Proceedings of the International Joint Conference on Neural Networks* (2012). doi:10.1109/IJCNN.2012.6252468.
- [4] D. Weimer, B. Scholz-Reiter, M. Shpitalni, Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection, *CIRP Annals Manufacturing Technology* 65 (1) (2016) 417–420. doi: 10.1016/j.cirp.2016.04.072. URL <http://dx.doi.org/10.1016/j.cirp.2016.04.072>.
- [5] Z. Yu, X. Wu, X. Gu, Fully convolutional networks for surface defect inspection in industrial environment, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10528 LNCS (October) (2017) 417–426.
- [6] H. Tian, F. Li, Autoencoder-based fabric defect detection with cross-patch similarity, *Proceedings of the 16th International Conference on Machine Vision Applications, MVA 2019* (2019). doi:10.23919/MVA.2019.8758051.
- [7] Wang, T., Chen, Y., Qiao, M., & Snoussi, H. (2018). A fast and robust convolutional neural network-based defect detection model in product quality control. *International Journal of Advanced Manufacturing Technology*, 94(9–12), 3465–3471. <https://doi.org/10.1007/s00170-017-0882-0>.
- [8] Y. Liu, M. M. Cheng, X. Hu, J. W. Bian, L. Zhang, X. Bai, J. Tang, Richer Convolutional Features for Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (8) (2019) 1939–1946. arXiv:1612.02103, doi:10.1109/TPAMI.2018.2878849.
- [9] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal Loss for Dense Object Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2) (2020) 318–327. arXiv:1708.02002, doi:10.1109/TPAMI.2018.2858826.
- [10] K. Oksuz, B. C. Cam, S. Kalkan, E. Akbas, Imbalance Problems in Object Detection: A Review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (10) (2021) 3388–3415. arXiv:1909.00169, doi:10.1109/TPAMI.2020.2981890.
- [11] D. Tabernik, S. Šela, J. Skvarč, D. Škočaj, Segmentation-based deep-learning approach for surface-defect detection, *Journal of Intelligent Manufacturing* 31 (3) (2020) 759–776. arXiv:1903.08536, doi:10.1007/s10845-019-01476-x.
- [12] J. Božič, D. Tabernik, D. Škočaj, Mixed supervision for surface-defect detection: From weakly to fully supervised learning, *Computers in Industry* 129 (2021). arXiv:2104.06064, doi: 10.1016/j.compind.2021.103459.
- [13] D. Racki, D. Tomazevic, D. Skocaj, A Compact Convolutional Neural Network for Textured Surface Anomaly Detection, *Proceedings 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018* 2018Janua (2018) 1331–1339. doi:10.1109/WACV.2018.00150.
- [14] L. Gautheron, A. Habrard, E. Morvant, M. Sebban, Metric learning from imbalanced data, *Proceedings International Conference on Tools with Artificial Intelligence, ICTAI 2019-November* (9) (2019) 923–930. arXiv:1909.01651, doi:10.1109/ICTAI.2019.00131.
- [15] S. Zhang, Z. Li, S. Yan, X. He, J. Sun, Distribution Alignment: A Unified Framework for Long-tail Visual Recognition (18) (2021) 1–10. arXiv:2103.16370. URL <http://arxiv.org/abs/2103.16370>
- [16] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique 16 (2002) 321–357.

- [17] H. Han, W.y. Wang, B.h. Mao, Borderline-SMOTE: A New Over-Sampling Method in (2005) 878–887.
- [18] A. Shrivastava, Training Region-based Object Detectors with Online Hard Example Mining arXiv: arXiv:1604.03540v1.
- [19] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem (2016) 770–778. arXiv:1512.03385, doi:10.1109/CVPR.2016.90.
- [20] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, J. Li, Dice Loss for Data-imbalanced NLP Tasks (2) (2020) 465–476. arXiv:1911.02855, doi: 10.18653/v1/2020.aclmain.45.
- [21] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2017). arXiv:1412.6980.
- [22] L. Lei, S. Sun, Y. Zhang, H. Liu, W. Xu, Psicnet: Pixel-wise segmentation and image-wise classification network for surface defects, *Machines* 9 (10) (2021). doi: 10.3390/machines9100221. URL <https://www.mdpi.com/2075-1702/9/10/221>.