ENSF619 Final Project: Small Object Detection: Steel Surface Defects

Zahra Arabi
Dept. of Electrical and Software Engineering
zahra.arabinarei@ucalgary.ca

Mariam Manzoor
Dept. of Electrical and Software Engineering
mariam.manzoor@ucalgary.ca

***Abstract*—The significance of identifying surface defects in steel cannot be overstated, as it plays a pivotal role in the industry. Traditional inspection methods have fallen short in both accuracy and efficiency. However, advancements in deep learning offer significant opportunities for improvement. This project addresses these limitations by focusing on refining the performance of the Faster R-CNN object detection model by integrating it with a Path Aggregation Network (PANet). This integration is designed to increase the capability of extracting features, by optimizing the flow of information throughout the feature hierarchy. This is achieved by incorporating bottom-up path enhancements, resulting in improving the depth and quality of feature extraction. Furthermore, in the post-processing phase, we opt for soft non-max suppression (Soft-NMS) over the conventional non-max suppression (NMS) approach. This adjustment leads to notable improvements in the recall and precision rates for various defect classes. The project aims to optimize results on the GC10–DET dataset, which comprises ten different types of defects at varying scales. This dataset poses a significant challenge due to the diversity of defect categories and the number of images. In conclusion, the methodologies applied in this project have successfully elevated defect detection performance, achieving a mean average precision (mAP) of 0.765. This is a significant improvement over the results obtained with the original Faster R-CNN algorithm. Thus, this approach demonstrates remarkable effectiveness in detecting small surface defects in steel, representing a significant advancement in the field.**

***Index Terms*—surface defect detection, object detection, deep learning, faster R-CNN**

I. Introduction

A surface defect's detection is of utmost importance in the steel industry, as it directly impacts both the integrity and safety of the final products. Surface anomalies such as scratches or cracks compromise the steel's structural integrity, diminishing its capacity to endure stress and pressure. These imperfections not only predispose the material to structural failures in applications ranging from architectural frameworks to critical pipeline infrastructures but also serve as initiation sites for corrosion processes. Such corrosion not only degrades the material further but can lead to catastrophic failures. Prompt identification of these surface defects is thus essential for initiating timely remedial measures, circumventing severe repercussions. Moreover, detecting these issues at an early stage prevents the progression of defective materials through the production line, thereby averting potential delays, resource wastage, and financial losses associated with product rejections. Consequently, accurate and expedient detection of surface imperfections is pivotal in upholding stringent quality standards, enhancing operational safety, curtailing unnecessary expenditures, and facilitating seamless production workflows within the steel manufacturing sector [1].

Traditional methods for object detection rely on extracting image features followed by the implementation of different algorithms to identify object profiles. These methods encompass edge detection algorithms such as the Sobel Operator [2] and Canny Edge Detector [3], along with techniques like Fourier Transform [4] and wavelet filtering [5]. However, these approaches are constrained by their limited capability for feature extraction, and the detection outcomes are notably susceptible to variations in lighting, pose, background clutter, noise, and other factors [6]. The advent of deep learning, particularly Convolutional Neural Networks (CNNs), has marked a significant advancement in addressing these constraints. CNNs are adept at learning features directly from the data, eliminating the need for manual feature specification. This capacity enables them to discern complex and subtle details within images, making them substantially more resilient to the variations commonly encountered in real-world settings.

On the other hand, detecting defects in steel presents its unique set of challenges. Firstly, there's a shortage of high-quality, real-world training datasets, and accurately labeling these datasets is equally challenging. Second, steel surfaces can be uneven or have imperfections like mill scale that can introduce noise and make defect detection more difficult. Thirdly, defects exhibit both inter-class similarity and intra-class diversity, meaning different types of defects can look alike while defects of the same type can appear very different. Moreover, various defect types may overlap on the steel surface, which can lead to some defects being overlooked, especially if they are detected with lower confidence [7].

In this project, we utilize the GC10-DET dataset [8], which includes ten types of defects sourced from real industrial environments. Our goal in training our model with this dataset is to boost its robustness, ensuring optimal performance in real-world scenarios. Additionally, to capture the complex features of defects, we employ the Faster R-CNN model [9] enhanced with the Path Aggregation Network (PANet) [10]. This integration occurs immediately following the Feature Pyramid Network (FPN) backbone [11]. PANet introduces a bottom-up path augmentation, enabling the upward flow of low-level features to refine the overall quality of the feature pyramid [10]. Implementing PANet has demonstrated improved performance on the GC10-DET dataset over the original Faster R-CNN framework. Lastly, in the post-processing phase, we adopt soft non-max suppression (Soft-NMS) [12] instead of the traditional non-maximum suppression (NMS) algorithm. While NMS tends to overlook overlapping object detection boxes due to its greedy nature, Soft-NMS merely lowers the confidence scores instead of outright discarding them. This subtle yet impactful change significantly boosts recall rates for the GC10-DET dataset. Our Codes and modules are available at: https://github.com/Mariam-73/ENSF-19-Final-Project

II. Related Works

With advancements in deep learning, numerous methods have been developed in the field of object detection to address the limitations of traditional approaches. Since the pioneering introduction of AlexNet [13], CNNs have been effectively applied to detect surface defects. In [14], a CNN framework was introduced for classification, localization, and object detection, utilizing both sliding window and multiscale approaches within the CNN framework. This model was also capable of learning to predict object bounding boxes, thereby localizing the object accurately. In [15], CNNs were being used for feature learning to detect rail surface defects, demonstrating the versatility of CNNs in various domains of object detection. In [16], research underscored the superiority of CNNs in computer vision tasks, particularly in segmenting and detecting surface anomalies. This study showcased CNNs' efficiency and effectiveness in complex vision tasks, as they demonstrated high accuracy with fewer parameters [8].

Surface defect detection algorithms leveraging object recognition are broadly categorized into single-stage and two-stage approaches. Single-stage algorithms, such as "You Only Look Once" (YOLO) [17], are designed for speed and efficiency, conducting object detection in a single pass across the image to simultaneously predict the class and location of potential defects. Their main advantage lies in their rapid processing capability, rendering them suitable for real-time detection applications. Additionally, single-stage algorithms are generally easier to implement and train than their two-stage counterparts. However, their primary drawback is potentially lower accuracy, especially in identifying small or overlapping defects. On the contrary, two-stage algorithms, exemplified by Faster R-CNN [9], prioritize accuracy through a two-step detection process: region proposal followed by object classification/localization. The initial phase utilizes a Region Proposal Network (RPN) to examine the image and propose potential object regions. Subsequently, these regions undergo meticulous refinement and classification in the second stage. By focusing computational efforts on image areas with a higher probability of containing defects, two-stage algorithms achieve enhanced precision in object detection, including the identification of surface anomalies. Two-stage algorithms excel in ensuring detection accuracy, especially in complex images containing multiple objects or defects. They can effectively handle objects of different sizes or overlapping ones. However, their complexity and longer processing times may restrict their applicability in situations requiring real-time detection, unless advanced hardware acceleration is available.

III. Materials and Methods

A. Dataset

In the study, we utilized the GC10-DET dataset which is available on GitHub[Link: https://github.com/lvxiaoming2019/GC10-DET-Metallic-Surface-Defect-Datasets]. The GC10-DET metallic surface defect dataset includes ten types of surface defects, i.e., punching (Pu), weld line (Wl), crescent gap (Cg), water spot (Ws), oil spot (Os), silk spot (Ss), inclusion (In), rolled pit (Rp), crease (Cr), and waist folding (Wf), as shown in "Fig. 1".

B. Detection Model

Our detection model, illustrated in "Fig. 2", is a Path Aggregation Network with MaxPooling based on the architecture of Faster R-CNN [9]. The model takes as input defect images resized to 256 by 128 pixels and produces bounding boxes along with corresponding defect labels as output in a two-step approach. In the first stage, a fully convolutional neural network proposes regions of interest (RoIs) where defects are likely to be situated. This is followed by a detector head to make the final predictions. The backbone of our detection model comprises a feature pyramid-based [11] ResNet18 [18] architecture, which is further enhanced with a bottom-up path to form a PANet [10]. Then, a Soft-NMS [12] module is employed to output the final detection results during model inference.

1) Feature Pyramid Network Backbone: FPN [11] was employed in our model, with ResNet18 [18] as the backbone convolutional architecture. In this framework, the FPN receives the input image and generates feature maps across multiple levels [11]. For the bottom-up pathway, we leverage a pre-trained ResNet18 [18] model, trained on the ImageNet-1K [19] dataset, with the last three layers configured for fine-tuning. Subsequently, the top-down pathway enriches the feature representation by up-sampling semantically strong feature maps from higher pyramid levels and merging them with bottom-up feature maps via lateral connections. This process yields feature maps that are both semantically and spatially stronger.

2) Bottom-up Path Augmentation: The FPN-based backbone [11] is augmented with a bottom-up path [10], which takes a higher-resolution feature map and merges it with the corresponding feature map from the FPN backbone network [11] using lateral connections. This process generates a new feature map for the augmented bottom-up structure. This addition to the backbone creates a pathway within the model, spanning from low-level feature maps to top-level feature maps, typically encompassing less than 10 layers [10] across these levels. Hence, generating a shortcut path facilitates the propagation of information from lower layers more effectively.

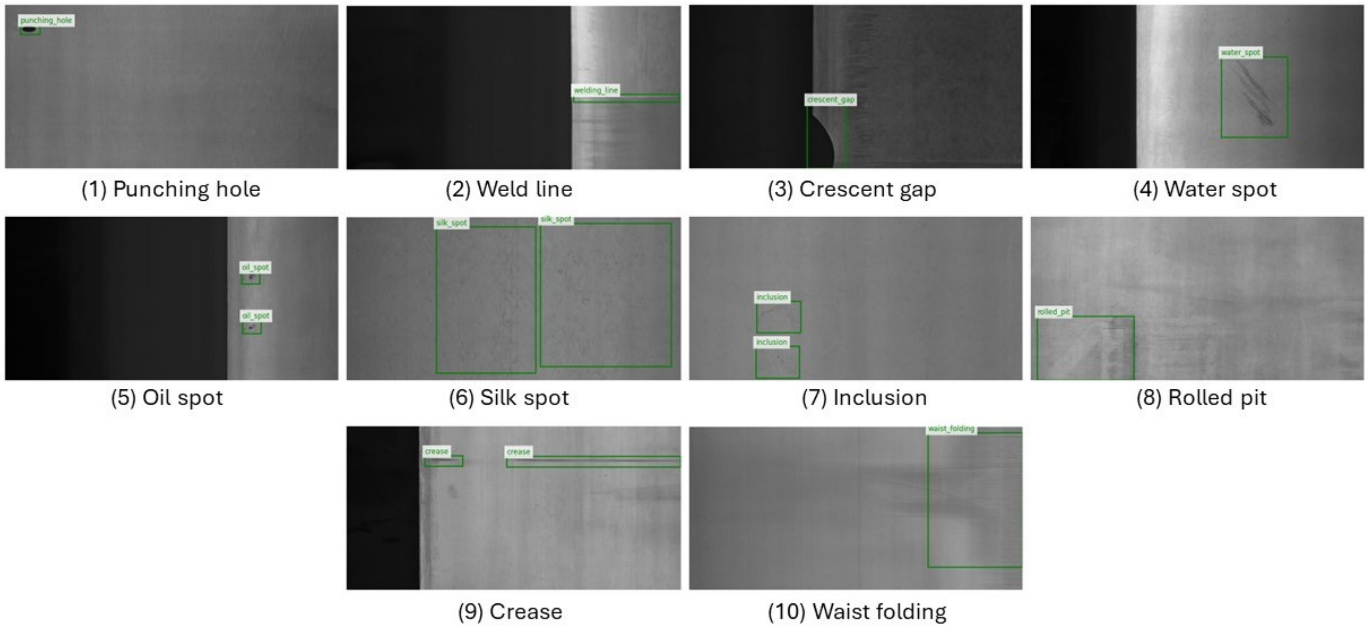

Fig. 1. Surface Defects in GC10-DET dataset. In sequence, the pictures are: (1)Punching, (2)Weld line, (3)Crescent gap, (4)Water spot, (5)Oil spot, (6)Silk spot, (7)Inclusion, (8)Rolled pit, (9)Crease, (10)Waist folding.

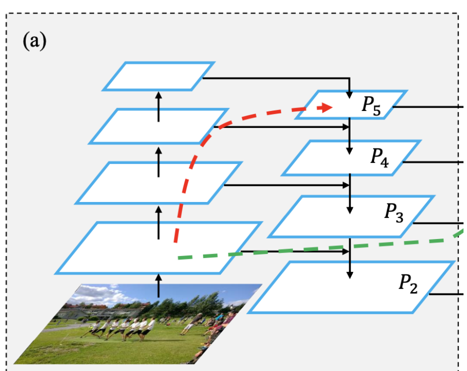

Fig. 2. Model Architecture.

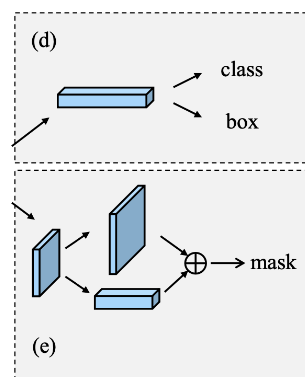

Fig. 3. Classification and Regressor Heads.

3) Region Proposal Network: The Faster R-CNN based model employs a RPN [9] to generate regions of interest. The RPN is a fully convolutional network designed to analyze images of any size and produce a series of rectangular object proposals, along with corresponding objectness scores. In our final model, we employed max pooling to reduce the spatial dimensions of feature maps before passing them to the fully connected detection head. This process creates candidate bounding boxes labeled as either objects or background, forming the input for a fully connected detection head.

4) Fully Connected Detection Head: The detection head, as shown in "Fig. 3", of the Faster R-CNN [9] based model is responsible for refining the regions of interest proposed by the RPN. It processes the extracted features from the proposed regions and outputs final predictions by refining the bounding boxes and classification of defects within those regions. By utilizing fully connected layers, classification, and regression heads, assigns defect labels and refines the bounding boxes.

5) Loss Function: The loss function used in training back propagates both classification and regression losses. For the RPN [9], there are two components: the classification loss (LRPNcls) and the regression loss (LRPNreg). The classification loss evaluates the model's ability to distinguish between objects and background, while the regression loss minimizes discrepancies between predicted and ground-truth bounding box coordinates. Similarly, for the detection heads, there are also two types of losses: a classification loss (Lheadcls) to categorize among the 10 defect classes, and a regression loss (Lheadreg) to refine bounding boxes. Therefore, the total loss function for both the RPN and detection heads is the sum of their respective classification and regression losses:

$$TotalLoss = L_{\text{RPN}_{cls}} + L_{\text{RPN}_{reg}} + L_{\text{head}_{cls}} + L_{\text{head}_{reg}} \quad (1)$$

IV. Results and Discussion

We conducted a series of experiments to assess the effectiveness of our Path Aggregation Network [10] with MaxPooling. Initially, we utilized Faster R-CNN [9] with ResNet18 [18] as our baseline model for basic detection tasks. To enhance defect detection within our dataset, which included multi-scale steel surface defects, we integrated FPN [11] to better identify smaller objects and augmented it with a bottom-up pathway to facilitate the propagation of crucial low-level features beneficial for identifying larger objects. The pooling layer of the Region Proposal Network [9] was further investigated, exploring adaptive feature pooling, average feature pooling, and max pooling techniques. Our experimental findings are presented through comprehensive visualizations and quantitative analyses.

1) Performance Evaluation: To evaluate our results, in the inference of our model, we used NMS and Soft-NMS [12] Techniques. We used the Recall, Average Precision (AP), and Mean Average Precision (mAP) of our model to measure our model's overall performance for each class.

2) Comparison Methods and Hyper Parameters: To benchmark our models against state-of-the-art detectors, we trained three variants on our dataset: Faster R-CNN based on ResNet18, Path Aggregation Network with Adaptive Feature Pooling, and Path Aggregation Network with Average Pooling. The final hyperparameters for each model included the Stochastic Gradient Descent (SGD) [20] optimizer with a scheduled learning rate of 0.001, momentum of 0.9, and weight decay of 0.0005.

3) Results and Discussion: "Table I" presents a detailed comparison of Recall results. The proposed method attains the highest recall scores for Ws, Os, Ss, In, Rp, Cr and Wf, and similar Recall for Pu and Gg. For Wl the recall scores marginally surpass those of our proposed model.

"Table II" shows detailed comparison results of Average Precision. The proposed method attains the highest AP in Cg, Ws, Os, Ss, In, Rp, and Wf. For Wl and Cr the AP scores were better than those of our proposed model, and for Pu, Faster R-CNN exhibited better recall scores.

TABLE I
Comparison of Recall on GC10-DET dataset

Defect Types	Faster R-CNN	PANet AvgPool	PANet Adaptive Feature Pooling	PANet MaxPool
Punching hole	0.96	0.96	0.92	0.96
Welding line	0.96	1.00	0.98	0.98
Crescent gap	1.00	1.00	1.00	1.00
Water spot	0.48	0.80	0.54	0.83
Oil spot	0.78	0.93	0.80	0.97
Silk spot	0.50	0.84	0.68	0.89
Inclusion	0.48	0.68	0.60	0.77
Rolled pit	0.23	0.84	0.61	0.92
Crease	0.25	0.50	0.25	0.75
Waist folding	1.00	0.85	0.44	1.00

"Table III" provides a comparison of our proposed model's inference using both NMS and Soft-NMS techniques and

TABLE II
Comparison of AP on GC10-DET dataset

Defect Types	Faster R-CNN	PANet AvgPool	PANet Adaptive Feature Pooling	PANet MaxPool
Punching hole	0.91	0.89	0.86	0.90
Welding line	0.88	1.00	0.83	0.97
Crescent gap	0.97	0.97	0.96	0.99
Water spot	0.28	0.73	0.43	0.76
Oil spot	0.59	0.88	0.66	0.96
Silk spot	0.23	0.46	0.31	0.60
Inclusion	0.25	0.49	0.25	0.55
Rolled pit	0.17	0.79	0.21	0.91
Crease	0.06	0.50	0.25	0.30
Waist folding	0.36	0.56	0.43	0.67
mAP	0.47	0.73	0.52	0.76

"Fig. 4" presents the Precision-Recall curve of our model. Our proposed model exhibits notable strengths in recall scores, indicative of its ability to effectively identify ground truth surface defects. However, the observed lower precision score, leading to a lower Average Precision, suggest a trade-off. This trade-off implies that while our model excels in detecting most objects of interest, it also tends to generate more false positives, highlighting areas for potential refinement in future iterations.

TABLE III
Comparison of AP and Recall for our proposed model using NMS and Soft-NMS

Defect Types	Recall		AP	
	NMS	Soft-NMS	NMS	Soft-NMS
Punching hole	0.96	0.96	0.90	0.90
Welding line	0.98	0.98	0.97	0.97
Crescent gap	0.96	1.00	0.96	0.99
Water spot	0.77	0.83	0.73	0.76
Oil spot	0.96	0.97	0.95	0.96
Silk spot	0.76	0.89	0.55	0.60
Inclusion	0.70	0.77	0.51	0.55
Rolled pit	1.00	0.92	0.94	0.91
Crease	0.75	0.75	0.43	0.30
Waist folding	0.85	1.00	0.65	0.67

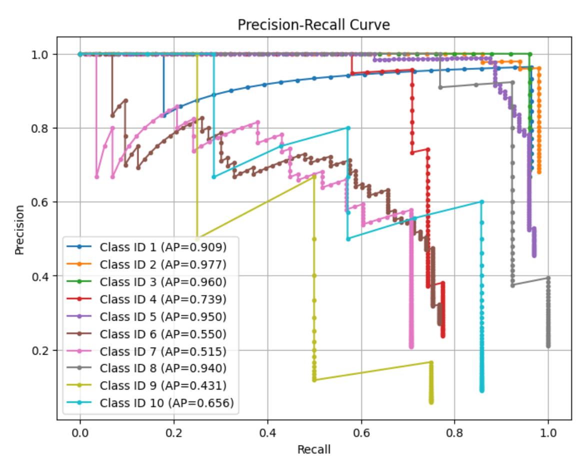

Fig. 4. Precision-Recall Curve

Moreover, “Fig. 5” depicts the training curve which highlights the robustness of our model, with minimal classification loss indicating precise and accurate classification performance. Such reliability in classification is critical in steel surface object detection, where the extent of defects dictates crucial decisions on resource allocation and intervention strategies.

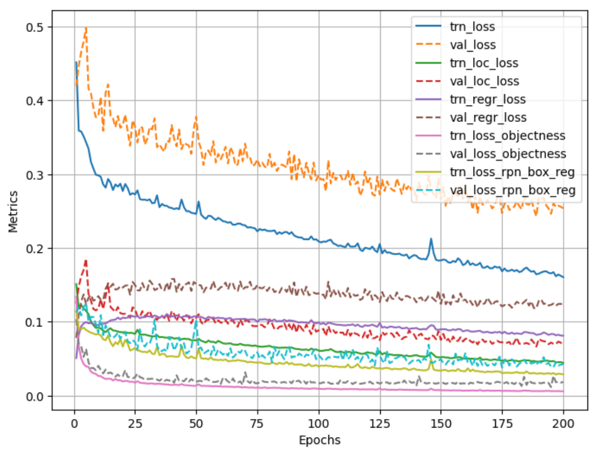

Fig. 5. Training and Validation Loss Curves of our proposed model

V. Conclusion

Our project leverages the GC10-DET dataset, containing ten industrial defects, aiming to improve our model’s effectiveness in real settings. We enhance detection accuracy by integrating the Faster R-CNN model with PANet, optimizing feature capture. Further, replacing traditional NMS with Soft-NMS boosts our model’s recall by refining for overlapping detections.

References

[1] S. Wang, X. Xia, L. Ye, and B. Yang, “Automatic Detection and Classification of Steel Surface Defect Using Deep Convolutional Neural Networks,” Metals, vol. 11, no. 3, p. 388, Feb. 2021, doi: https://doi.org/10.3390/met11030388.

[2] N. Kanopoulos, N. Vasanthavada and R. L. Baker, ”Design of an image edge detection filter using the Sobel operator,” in IEEE Journal of Solid-State Circuits, vol. 23, no. 2, pp. 358-367, April 1988, doi: 10.1109/4.996.

[3] J. Canny, “A Computational Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, doi: https://doi.org/10.1109/tpami.1986.4767851.

[4] R. Stojanovic, P. Mitropulos, C. Koulamas, Y. Karayiannis, S. Koubias, and G. Papadopoulos, “Real-Time Vision-Based System for Textile Fabric Inspection,” Real-Time Imaging, vol. 7, no. 6, pp. 507–518, Dec. 2001, doi: https://doi.org/10.1006/rtim.2001.0231.

[5] X. Li, S. K. Tso, X. -P. Guan and Q. Huang, ”Improving Automatic Detection of Defects in Castings by Applying Wavelet Technique,” in IEEE Transactions on Industrial Electronics, vol. 53, no. 6, pp. 1927-1934, Dec. 2006, doi: 10.1109/TIE.2006.885448.

[6] H. Wang, M. Li, and Z. Wan, “Rail surface defect detection based on improved Mask R-CNN,” Computers and Electrical Engineering, vol. 102, p. 108269, Sep. 2022, doi: https://doi.org/10.1016/j.compeleceng.2022.108269.

[7] X. Wen, J. Shan, Y. He, and K. Song, “Steel Surface Defect Recognition: A Survey,” vol. 13, no. 1, pp. 17–17, Dec. 2022, doi: https://doi.org/10.3390/coatings13010017.

[8] X. Lv, F. Duan, J. Jiang, X. Fu, and L. Gan, “Deep Metallic Surface Defect Detection: The New Benchmark and Detection Network,” Sensors, vol. 20, no. 6, p. 1562, Mar. 2020, doi: https://doi.org/10.3390/s20061562.

[9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: https://doi.org/10.1109/tpami.2016.2577031.

[10] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path Aggregation Network for Instance Segmentation,” arXiv:1803.01534 [cs], Sep. 2018, Available: https://arxiv.org/abs/1803.01534.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” arXiv:1612.03144 [cs], Apr. 2017, Available: https://arxiv.org/abs/1612.03144.

[12] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS – Improving Object Detection With One Line of Code,” arXiv:1704.04503 [cs], Aug. 2017, Available: https://arxiv.org/abs/1704.04503.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2012, doi: https://doi.org/10.1145/3065386.

[14] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,” arXiv:1312.6229 [cs], Feb. 2014, Available: https://arxiv.org/abs/1312.6229.

[15] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska and B. De Schutter, ”Deep convolutional neural networks for detection of rail surface defects,” 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 2016, pp. 2584-2589, doi: 10.1109/IJCNN.2016.7727522.

[16] D. Racki, D. Tomazevic and D. Skocaj, ”A Compact Convolutional Neural Network for Textured Surface Anomaly Detection,” 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 2018, pp. 1331-1339, doi: 10.1109/WACV.2018.00150.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” arXiv.org, Jun. 08, 2015. https://arxiv.org/abs/1506.02640.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” arXiv.org, Dec. 10, 2015. https://arxiv.org/abs/1512.03385.

[19] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, ”ImageNet: A large-scale hierarchical image database,” 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[20] S. Ruder, “An overview of gradient descent optimization algorithms *,” Jun. 2017. Available: https://arxiv.org/pdf/1609.04747.pdf.