Available online at www.eccomasproceedia.org
Eccomas Proceedia SMART (2023) 1209-1220

ECCOMAS
Proceedia

X ECCOMAS Thematic Conference on Smart Structures and Materials
SMART 2023
D.A. Saravanos, A. Benjeddou, N. Chrysochoidis and T. Theodosiou (Eds)

CENTERNET-BASED MODELS FOR THE DETECTION OF DEFECTS IN AN INDUSTRIAL ANTENNA ASSEMBLY PROCESS

THEODOSIOS C. THEODOSIOU[*], T. TZIOLAS[*], K. PAPAGEORGIOU[*], A. RAPTI[*], E. PAPAGEORGIOU[*], S. PANTOJA[†], P. CHARALAMPOUS[‡], N. DIMITRIOU[‡], D. TZOVARAS[‡], A. CUIÑAS[†], J. MOURELLE[†], ANDREAS BÖTTINGER[⁂], G. MARGIETIS[‼]

[*] Dept. of Energy Systems
University of Thessaly, Gaiopolis Campus,
GR-41500 Larissa, Greece, e-mail: dozius@uth.gr

[†] Televes, Rúa B. de Conxo, 17
Santiago de Compostela, 15706, A Coruña, España

[‡] Centre for Research and Technology Hellas,
Information Technologies Institute,
6th Km Charilaou-Thermi Road, 57001 Thermi-Thessaloniki, Greece

[⁂]EyeVision Technology (EVT)
Ettlinger Str. 59, 76137 Karlsruhe, Germany

[‼]Foundation for Research and Technology Hellas, Institute of Computer Science,
Heraklion, Crete, 70013, Greece

Abstract. This work presents a study for the identification of incorrect antenna assemblies using Artificial Intelligence. The anchor-free and lightweight object detection CenterNets are combined with different feature extractors and their performance is assessed regarding the trade-off between high accuracy and low latency, as applied on a real industrial dataset with limited defective-sample images. To enhance the training dataset, synthesized defects are induced into the dataset, followed by heavy data augmentation which is performed on-the-fly during training. In the proposed methodology the classification loss is further penalized compared to localization loss, aiming at industrial customization, in which finding defects and minimizing False Positives are of high importance. The CenterNet-ResNet50 architecture yields superior performance. The performance comparison against other state-of-the-art computer vision classifiers proves that the proposed detection approach is more efficient than the typical binary classification with transfer learning. The latency of the proposed detector is

Peer-review under responsibility of the organizing committee of SMART 2023.
doi: 10.7712/150123.9869.444634

estimated for both GPU and CPU inference to pave the wave for real-time defect detection applications.

Key words: Defect Detection, CenterNet, Antenna Industry, Assembly Process, Transfer Learning.

1 INTRODUCTION

Modern manufacturing assembly lines excel at processing disparate materials with high accuracy, yet the utter avoidance of defects is rather impossible. The complex nature of these procedures generates defects that often are propagated and amplified through process stages [1]. Thorough quality inspection throughout the manufacturing cycle of products is vital for ensuring low-cost and high-quality production. Automatic quality inspection systems have emerged to supersede manual inspections, as their performance is faster and more accurate in large scale productions. These systems employ machine vision cameras and exploit the advantages of the AI-based methods to meet the requirements of smart manufacturing.

Beyond the straightforward classification of products into healthy and defective with computer vision algorithms, an enhancement can be accomplished with an accurate estimation of the location of the defect. This holistic defect detection approach is advantageous in the decision-making process that the operator follows, as it guides them towards the location that "triggered" the detection. Defect detection falls in the scope of object detection, which is not a trivial computer vision task. The convolutional object detectors in literature are distinguished in a) single-stage and b) two-stage [2]. The first category involves models that can perform object detection in a single shot and are preferred for real time detection thanks to their fast detection capacity. The second category requires a "region proposal module" at the first stage and a "consecutive region classifier network" at the second stage. Due to the two-stage nature, they are slower, yet more accurate models.

In general, all object detection algorithms are trained on massive datasets such as COCO [3] and their performance is assessed based on predefined metrics, such as the mean Average Precision (mAP) and the inference speed. The primary metrics are focused on the localization performance and fail to provide insights for the False Positives (FP). The object classes in the pictures of COCO datasets are dense, with low variation in shapes and with high overlap area between them, whereas defects in products may possess unique characteristics. Therefore, defect detection requires a more sophisticated approach to be effectual in real industrial use cases. It is proposed that additional indicators should be considered, including the Area of Overlap or the Intersection over Union (IoU) threshold, the anchor boxes, and the sensitivity to false alarms.

Anchors are predefined bounding boxes that are employed by object detection algorithms as proposals to the final bounding boxes. Their definition poses a challenge for the accurate defect detection approach. During training, the model learns to match one of several pre-defined anchor boxes to the ground truth bounding boxes. The use of clustering methods has been proposed in literature to improve the estimation of the proper anchors [4]. In industrial cases, however, the inconsistency of defect shapes and the scarcity of the defected samples cause difficulties in anchor definition.

In this work, an anchor-free defect detection scheme is assessed for a distinct stage of the robotized antenna assembly line of Televes S.A.U within the scope of OPTIMAI [5]. Pre-trained lightweight object detection models are fine-tuned to detect breaks and housing imperfections in the antenna reflectors, caused by incorrect folding and/or insertion of elements. The TensorFlow Object Detection API (TFOD) [6] model repository[1] was empolyed to acquire effective pre-trained models for this task. The CenterNets [7] are state-of-the-art, lightweight, and anchor-free models that offer an optimal trade-off between accuracy and inference speed.

The contribution of this work is summarized as:

1. Pre-trained CenterNets with different backbone networks are assessed with transfer learning in a real industrial dataset. They are compared to the straightforward binary classification with transfer learning to demonstrate the effectiveness of this approach.
2. Contrary to standard object detection approaches, where localization is further rewarded, in the employed methodology lower IoU threshold is allowed for the proposal penalization during training. In addition, higher weight is passed to the classification loss task. Thus, by allowing localization variations, the AP is increased from 0.88 (at IoU 0.5), to 0.94 (at IoU 0.3) and more complex defect shapes can be detected. Furthermore, the performance in both False Positives (FP) and False Negatives (FN) is assessed at each confidence threshold with the F1-Score to further evaluate the false alarms in the production line.
3. To mitigate data scarcity and to ensure proper training without overfitting, synthetic defective samples are produced, and heavy data augmentation techniques are employed.

The rest of this paper is organized as follows. Section 2 provides the related work in defect detection, Section 3 describes the methodology of this work, Section 4 presents and discusses the results, and Section 5 presents the conclusions derived from this work.

2 RELATED WORK

A thorough literature survey was conducted in popular research databases like Google Scholar, using formal search queries, such as “antenna AND (assembly OR manufacturing) AND defect detection”. Results support the existence of multiple deep learning defect detection approaches [2], but no prior work has been found in the area of detecting assembly errors using the CenterNets. Most works are focused on sensors and non-destructive testing techniques [8] which is a totally different area. To this end, this section is focused on recent findings regarding defect detection with special focus on CenterNets.

A CenterNet with distillation knowledge, namely D-CenterNet, was proposed in [9] for fabric defect detection. Distillation knowledge is used to train a smaller network with a larger and more accurate model imitating a teacher-student scheme. In this implementation, the backbone was modified by integrating a strip pooling module to improve feature extraction of defects with extreme aspect ratios and background complexity. The authors evaluated their methodology in two open datasets, the Aliyun Tianchi fabric dataset and the NEU-DET hot-rolled steel defect detection dataset. AP of 75% was achieved for the fabric dataset, which is a 3% improvement compared to the second best model. A CenterNet variation, termed as DCC-

[1] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

CenterNet, was employed in [10] to further improve the accuracy in the NEU-DET steel defect dataset. By modifying the parameters of the network and the loss function, mAP 79% was achieved.

CenterNets were thoroughly examined in [11] to assess their ability to detect surface defects in Additive Manufacturing. CenterNets with different backbone networks, hyperparameters and datasets were investigated. A modification was proposed in the loss function with the addition of Count Loss targeting performance increase. The derived model achieved AP of 50% at IoU 0.5 and surpassed previous works in the domain that involved both CenterNets and other popular detectors.

A lightweight CenterNet was employed in the [12] for the detection of defects in high voltage insulators. The lightweight MobileNetV1 network was employed as a feature extractor and a channel attention module in the CenterNet to improve the detection abilities. This detector was tested in both a public and a custom insulator dataset, and improved the detection accuracy by 12%. In addition, the architecture of the MobileNetV1 reduced the required inference time.

3 METHODOLOGY

3.1 CenterNet

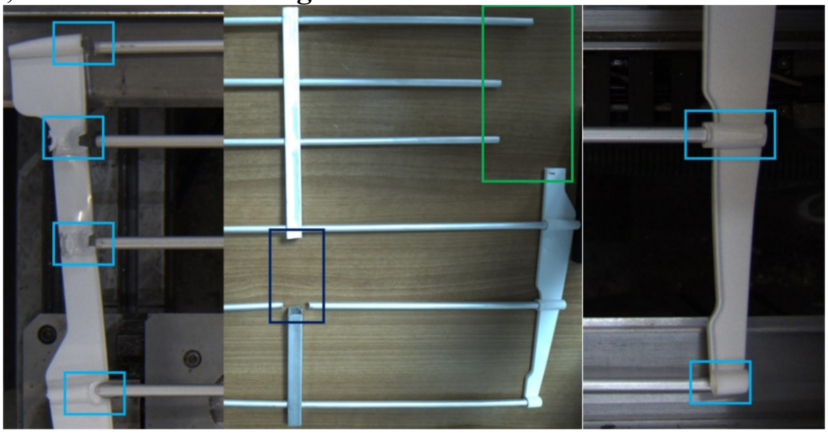
CenterNet is a single-stage model, yet it provides abilities of two-stage detectors without the additional computational cost. The CenterNet detects each object as a triplet of keypoints, namely the “center”, the “top left corner” and “bottom right corner”. As the name suggests CenterNet focuses on the center information. The original implementation uses the Hourglass [13] as its backbone network which has repeated bottom-up and top-down processing layers. Its developers proposed two modules to achieve higher detection accuracy of the keypoints: 1) the center pooling, and 2) the cascade corner pooling. These modules cast the visual patterns in objects into a keypoint detection process. The architecture has three output heads and one loss per head:

1. The heatmap. This has channel depth equal to the number of classes and determines the object center based on the peak values. The heatmap employs the focal loss [14] which is a modified version of cross-entropy loss; focal loss performs well with class imbalance.
2. The object size. This estimates the size of an object. It employes the L1 loss.
3. The local offset. This aims to correct the center point prediction and also employs the L1 loss.

The selection of a backbone network is essential as it directly affects memory needs, inference speed, and detection capabilities. In this work, the lightest versions of the CenterNet were examined, which engaged the ResNet50, the ResNet101 [15] and the MobileNetV2 [16] as feature extraction networks. ResNets employ residual blocks or the technique known as “skip connections” in the architecture of the Convolutional Neural Network (CNN), in order to solve the vanishing/exploding gradient problem in deep networks. MobileNetV2 on the other hand, is a CNN network that was designed to be efficient with limited resources, such as edge and mobile devices.

3.2 Dataset

The area scan camera FLIR Blackfly S BFS-PGE-200S6C was installed on the production line to monitor the assembly process and to acquire a suitable dataset for defect detection. The sensor captured images at resolution 4401×2898 in RGB format. The production line was optimally calibrated, which resulted in very limited defective samples. After inducing additional images from lab measurements, the dataset contained 165 images of defective and >4000 image of healthy antennas. Training of TFOD does not accept hard negatives (i.e. images with no object of interest), thus, the images of healthy antennas were used only for testing. Three defect classes were defined by experts: 1) housing imperfection, 2) metal crack, and 3) plastic crack; these are shown in **Figure 1**.



(a) (b) (c)

Figure 1: Indicative defects belonging to each one of the three classes, as defined by experts. (a) housing imperfections, (b) metal cracks, (c) plastic cracks.

The dataset was split into training, validation and testing sub-datasets. Due to the unbalance in the defect classes, the training and validation datasets were further enriched with synthetic samples, using the following simple process: First, a healthy and a defective sample image were overlaid and aligned. Then, different areas of the healthy sample image were systematically erased to simulate actual defects (**Figure 2**). To prevent bias, only a limited number of new images were produced.

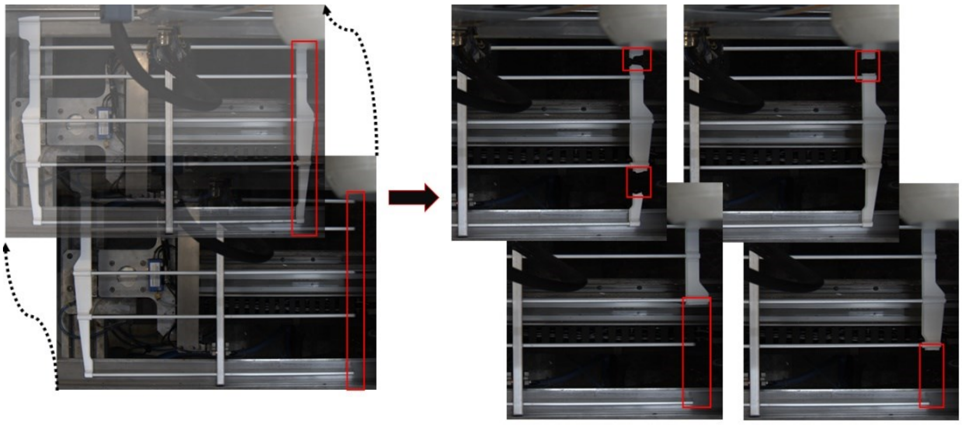


Figure 2: Production of synthetic defective-antenna images. Left: Overlaying of a healthy- and a defective-antenna image. Right: Synthetic defective-antenna images.

3.3 Model Pipeline and Configuration

All the preprocessing and the postprocessing steps were performed on-the-fly using the configuration files of the pre-trained TFOD models. These configuration files control the whole pipeline and parametrization of the model. Initially, the configuration of the preprocessing steps was considered. Images were resized to 512×512, following the requirements of CenterNets. Since the resizing was performing on-the-fly, the zero-padding method was preferred, as it provides a time efficient approach without compromise in accuracy [17]. Subsequently, a heavy data augmentation pipeline was employed to artificially increase the diversity of the training dataset by:

1. Geometrical transformations: horizontal and/or vertical flipping, cropping and/or scaling of images with randomness.
2. Color space transformations: random RGB to gray transformations, random adjustments in hue/contrast/brightness/saturation.
3. Noise insertion: random patches of gaussian noise, random jpeg quality.
4. Mixing images: random self-concatenation in both the vertical and horizontal axes.

Example images after data augmentation are shown in **Figure 3**.

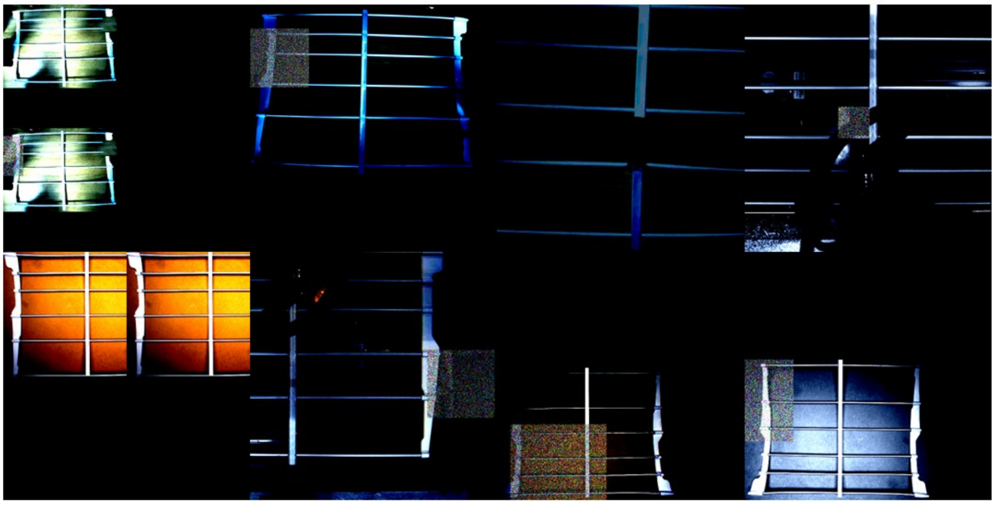


Figure 3: Indicative images produced using data augmentation to enhance the training process.

The feature extraction backbone network was considered to guide the parametrization and fine-tuning of the model. Performance comparisons were performed against the four different feature extractors provided by the CenterNets. The total output classes in the classification head were set to three, to match as the expert-defined defect classes. As the total number of defects in each image never exceed 10, the requested number of predicted boxes (defects) was set to 20. This value ensures computational efficiency against the default value (100) and leaves a margin for future developments.

The well-known Adam [18] optimizer was employed for training, following the original implementation of the network. Due to memory limitations, the learning rate parameters were modified. The cosine decaying learning rate was utilized with learning rate base from 0.001 to 0.0001 and warm up learning rate from $2.5 \cdot 10^{-4}$ to $2.5 \cdot 10^{-5}$. The batch size for each backbone architecture was different. For HourGlass104, the batch size was set to 2; for ResNet101, it was set to 4; for ResNet50, it was set to 8; and for MobileNetV2, it was set to 16. Although the default loss functions were employed for all models, parameters were modified. Increased weight (× 1.2) was passed to the task loss, and decreased threshold (0.6) was used in Intersection over Union (IoU), Eq. (1), for filtering predictions. This led to an approach focused more on defect classification accuracy rather than localization accuracy. Moreover, the COCO metrics were exploited only during training and the performance in the test dataset was assessed with the F1-Score, the Average Precision (AP), as defined in Eq. (2-3) respectively, at the lowest COCO IoU threshold (0.5). The total training steps were set to 70,000 and the checkpoint callback was employed to store the model weights every 2,000 steps. Each checkpoint was further assessed within the TensorBoard to obtain the best weights derived during training.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{Predicted\ \cap Ground\ Truth}{Predicted\ \cup Ground\ Truth} \tag{1}$$

$$F1\text{-}Score = 2\times (Precision \times Recall) / (Precision + Recall) \quad (2)$$

$$AP = \int_0^1 Precision(recall)\, d(recall) \quad (3)$$

4 RESULTS AND DISCUSSION

The experiments were conducted on a computational workstation with Intel Core i9-11900KF @ 3.5GHz CPU, 16 GB RAM and NVIDIA GeForce RTX 3080 Ti with 12GB of GDDR6X memory, running Windows 10 Professional. In-house code was developed in Python, employing TFOD API (TensorFlow v.2.9.1). Each CenterNet model was combined with a different feature extraction network. Five different instances of each model were trained, and the best metrics are presented in **Table 1**. These metrics attained at the presented confidences thresholds, which were further employed to filter out detections. In essence, at high confidence scores, a model is expected to produce more TPs than FPs, while as the confidence score decreases more FPs are expected. Therefore, by ranking all detections in the unseen test dataset in a descending order of their confidence scores, the thresholds that produced the maximum F1-Score were found. The latency was calculated for both GPU and CPU processing, considering the potential hardware limitations in an industrial implementation. Moreover, the "average time per training step" is provided as an indicator for training duration which depends on the batch size and the architecture of the network.

Results suggest that the highest performance was achieved by the CenterNet-ResNet50 as its feature extractor network, whereas the slightly deeper CenterNet-ResNet101 performed almost equally well. Moreover, the CenterNet-ResNet50 presented the highest AP of 0.88 and F1-Score of 0.89 which further supports its detection capabilities with high Precision and high Recall. In addition, it exhibited the second lowest latency after the CenterNet-MobileNetV2 which was the fastest model in terms of inference speed. Interestingly, the CenterNet-HourGlass104 achieved the worst performance in all metrics for this case study, despite being one of the most accurate models in literature.

Table 1: Performance metrics of the CenteNets

Model	AP IoU 0.5	F1-Score IoU 0.5	Confidence threshold	GPU Latency (ms)	CPU Latency (s)	Time per step (s)
CenteNet MobileNetV2	0.8	0.86	0.21	26	0.17	3.1
CenterNet ResNet50	0.88	0.89	0.45	28	0.19	2.9
CenterNet ResNet101	0.86	0.87	0.56	40	0.33	1.3
CenterNet HourGlass104	0.8	0.85	0.45	93	0.83	0.7

Furthermore, a performance comparison was examined with pre-trained on Imagenet [19] CNN classifiers. The binary classification is a straightforward choice to distinguish healthy and defective products; thus, the pre-trained backbone networks of the CenterNets with the highest performance (**Table 1**) were assessed for this task. The ResNet50, ResNet101 and MobileNetV2 were modified to have a) input dimension 512×512, b) an average global pooling layer after the final convolution layer, c) a dropout layer with 0.2 probability for regularization and d) a single output head for binary classification. In contrast to the CenterNet detectors, which were trained only on images of defective samples, the classifiers were trained on both healthy and defective antennas. In fact, the number of healthy samples was twice the size of the defected to represent the imbalance of real industrial datasets. To alleviate the imbalance, higher weight (×1.2) was passed to the defected samples. The training was performed in two steps. First, the classification head was trained and subsequently the whole network was fine-tuned with low learning rate (10^{-6}) in Adam optimizer. The detection assessment was modified as well to match the binary classification form (i.e., healthy if no defect was detected or defective if one or more defects were detected). Results are shown in **Table 2**.

Table 2: Performance comparison for the binary classification task.

Model	Accuracy
MobileNetV2	0.72
ResNet50	0.77
ResNet101	0.79
CenterNet-MobileNetV2	0.95
CenterNet-ResNet50	0.95
CenterNet-ResNet101	0.93

It is noticed that the proposed defect detection approach managed to surpass the classic binary classification approach. Due to limited samples, the classifiers struggled with learning the differences between healthy and defective samples. In contrast, by pointing out the defects and due to the existence of more than one defect in most of the defective samples, the detection approach proved to be far more efficient. It is envisioned that the performance of the classifiers could be further enhanced by inserting fully connected layers, but the additional complexity of such an architecture deserves a paper of its own.

Finally, the detection could be improved by modifying the IoU threshold. As previously stated, localization can be allowed to be less robust, due to the inconsistency of defect shapes and sizes. As an example, **Figure 4** demonstrates predictions for a plastic crack defect with a complex shape. The IoU values between the ground truth (red dashed line) and predicted bounding boxes (green solid lines) suggest that both detections were FPs due to low IoU (<0.5). Although, the CenterNet-ResNet50 failed to provide accurate localization, it correctly detected the presence of a defect in that area, which could guide a human operator to resolving this issue. For this case study, lowering the IoU threshold down to 0.3 was found to yield good results. In fact, the CenterNet-ResNet50 yielded 0.94 F1-Score and 0.94 AP at 0.3 IoU, which can be considered as a 6% performance improvement.

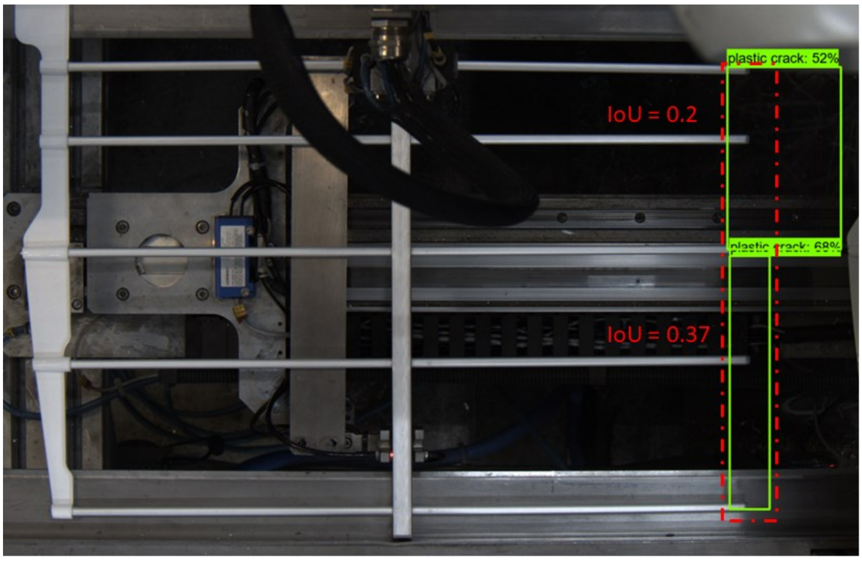


Figure 4: Example case study to demonstrate the effect of lowering the IoU threshold.

5 SUMMARY AND CONCLUSIONS

The importance of early defect detection in manufacturing and assembly lines is well-known. Various AI-based methods have been deployed, involving machine vision and classification. This work presented an AI-based method for the efficient detection of defects in a real industrial line, during the assembly stage of antennas. An area scan camera was used to monitor the process and capture a series of images. Since the number of defective samples was very limited, synthetic images and laboratory measurements were incorporated to alleviate data scarcity. Various CenterNet-based architectures were assessed for detecting defects in the assembly of antennas. Each CenterNet architecture integrated a different backbone network for feature extraction, namely the HourGlass104, the MobileNetV2, the ResNet50 and the ResNet101 networks. It was concluded that the CenterNet-ResNet50 was the most efficient model for this task, as it achieved AP 0.88 and F1-Score 0.89 at 0.5 IoU. Apart from the superior detection performance, this architecture required an average of only 28ms for GPU inference and 190s for CPU inference. The proposed architectures proved to be more efficient than the classic binary classification transfer learning, as the typical ResNet50, ResNet101 and MobileNetV2 classifiers failed to efficiently learn the structure differences between healthy and defective samples. Additional investigation demonstrated that a decrease of the IoU filtering to 0.3 affected both AP and F1-Score by increasing their values to 0.94. Future work will focus on modifications of the architecture (e.g., inserting fully connected layers), and processing of larger datasets.

ACKNOWLEDGEMENT

This work has been supported by the EU Project OPTIMAI (H2020-NMBP-TR-IND-2020-singlestage, Topic: DT-FOF-11-2020, GA 958264). The authors acknowledge this support.

REFERENCES

[1] X. Wang, M. Liu, M. Ge, L. Ling, and C. Liu, “Research on assembly quality adaptive control system for complex mechanical products assembly process under uncertainty,” *Computers in Industry*, vol. 74, pp. 43–57, 2015, doi: https://doi.org/10.1016/j.compind.2015.09.001.

[2] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, “A survey of modern deep learning based object detection models,” *Digital Signal Processing*, vol. 126, p. 103514, Jun. 2022, doi: 10.1016/j.dsp.2022.103514.

[3] T.-Y. Lin *et al.*, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[4] Y. Zhong, J. Wang, J. Peng, and L. Zhang, “Anchor box optimization for object detection,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1286–1294.

[5] “OPTIMAI,” *EU Project GA 958264*. https://optimai.eu/

[6] J. Huang *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.

[7] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as Points.” arXiv, Apr. 25, 2019. Accessed: Dec. 28, 2022. [Online]. Available: http://arxiv.org/abs/1904.07850

[8] M. Peng, D. Wang, L. Liu, Z. Shi, J. Shen, and F. Ma, “Recent Advances in the GPR Detection of Grouting Defects behind Shield Tunnel Segments,” *Remote Sensing*, vol. 13, no. 22, 2021, doi: 10.3390/rs13224596.

[9] Z. Liu, W. Lyu, C. Wang, Q. Guo, D. Zhou, and W. Xu, “D-CenterNet: An Anchor-Free Detector With Knowledge Distillation for Industrial Defect Detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022, doi: 10.1109/TIM.2022.3204332.

[10] R. Tian and M. Jia, “DCC-CenterNet: A rapid detection method for steel surface defects,” *Measurement*, vol. 187, p. 110211, 2022, doi: https://doi.org/10.1016/j.measurement.2021.110211.

[11] R. Wang and C. F. Cheung, “CenterNet-based defect detection for additive manufacturing,” *Expert Systems with Applications*, vol. 188, p. 116000, 2022.

[12] H. Xia, B. Yang, Y. Li, and B. Wang, “An Improved CenterNet Model for Insulator Defect Detection Using Aerial Imagery,” *Sensors*, vol. 22, no. 8, 2022, doi: 10.3390/s22082850.

[13] A. Newell, K. Yang, and J. Deng, “Stacked Hourglass Networks for Human Pose Estimation.” arXiv, 2016. doi: 10.48550/ARXIV.1603.06937.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016,

pp. 770–778.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” 2018, doi: 10.48550/ARXIV.1801.04381.

[17] M. Hashemi, “Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation,” *Journal of Big Data*, vol. 6, no. 1, p. 98, Nov. 2019, doi: 10.1186/s40537-019-0263-7.

[18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.