Available online at www.eccomasproceedia.org
Eccomas Proceedia SMART (2023) 1270-1278

ECCOMAS
Proceedia

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

EFFICIENTDET APPLICATION FOR DETECTION OF INCORRECT ASSEMBLIES IN THE ANTENNA MANUFACTURING PROCESS

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

[*] Dept. of Energy Systems
University of Thessaly, Gaiopolis Campus,
GR-41500 Larissa, Greece, e-mail: elpinikipapageorgiou@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 aims to investigate and effectively apply the new deep learning methods for object detection from the family of EfficientDet-Lite algorithms for the detection of incorrect assemblies in the antenna manufacturing process. In the proposed methodology, state-of-the-art pre-trained models are exploited and fine-tuned in a real industrial dataset with limited samples. The defects considered in the dataset include metal cracks, plastic cracks, and housing imperfection. The proposed approach with EfficientDet-Lite2 can achieve an Average Precision (AP50) of 73.68%. A Yolov5-based model was also developed and extracted comparable results. Moreover, the Average Precision for each class was computed. For conducting the experiments, the TensorFlow Lite object detection API was employed. The results of this research work are promising and support further investigation of these types of deep learning for defect detection in manufacturing.

Keywords: Defect Detection, Smart Structures, Assembly Process, EfficientDet, Deep Learning, Object Detection.

doi: 10.7712/150123.9874.444639

1 INTRODUCTION

In the manufacturing process, assembly operations are critical and require extreme caution [1]. During the manufacturing production line, the assembly of materials for the construction of a complete object is the main assignment. Regarding the antenna assembling, metallic source reflectors are connected with plastic, for the production of the final product. Even with cutting-edge equipment, errors still occur in the manufacturing process. One of the main reasons for a defective part is that the process is quite complicated, and it depends on the quality of each individual material. For this reason, proper quality control is essential.

Manual inspection of the whole process, including quality checks on each material is extremely deficient and expensive, thus it is not applicable for large-scale manufacturing processes. For this reason, automated processes, based on computer vision methodologies, have been developed and properly adapted, and offer new capabilities for visual inspection automatically, efficiently and in less time.

Visual quality inspection is related to image classification, which practically recognizes data patterns from numerous different categories. Object detection is also relevant to image classification, with the addition of evaluating the position and the size of an object, utilizing a bounding box. Object detection can be considered as an advanced image classification approach for defect detection, since it can be directed to identify defects along with interpretation.

Literature review revealed numerous defect detectors. The most popular detectors for real-time applications, i.e. for applications where fast inference is crucial, are the YOLO family [2], [3], EfficientDet [4], CenterNet [5], etc. Recently, transformer-based models were proposed, such as Visual Transformer [6], that inherit concepts previously applied to natural language processing and are modified for object detection.

In [15] the authors addressed aluminum casting with the development of EfficientDet and YOLO. Due to the lack of defective samples, the researchers generated simulated ellipsoidal defects onto the X-ray training images. Regarding the original images, they were used for the testing procedure. They experimented with different thresholds for Intersection over Union (IoU), to evaluate the model's ability to detect small defects. According to the results, Yolov5 achieved the highest mAP with a value of 0.90 utilizing an IOU of 0.25. The training duration we 2.5 hours.

In [16] EfficientDet was exploited for ultrasonic material inspection. The researchers experimented with the versions of D0-D2 of EfficientDet and compared the results with other state-of-the-art detectors. The dataset contained over 4000 B-scan images, with 68 unique defects. The results demonstrated that EfficientDet-D0 achieved the best mAP with a value of 89.6%. Moreover, 5-fold cross-validation was utilized. The authors proposed the usage of the K-means algorithm for the calculation of anchors, which has been adopted in the recent versions of YOLO for improving mAP.

In [17] the authors developed EfficientDet-D0 to address fabric defect detection. The authors tested five different fabric datasets and utilized data augmentation. K-means clustering was utilized for the shapes of the anchor boxes. The proposed model achieved over 90% mAP. It was also tested with NVIDIA TensorRT, where the authors proved that the computation time can be 2.5 times faster than the cloud-based method for reasons related to data transmission latency.

It is worth mentioning that real industrial data are often scarce or imbalanced [7]. This poses a challenge in the development of robust solutions with deep learning, and hence transfer learning and data augmentation techniques have been employed in the literature. TensorFlow Object Detection (TFOD) offers an Application Programming Interface (API) [8] which contains state-of-the-art pre-trained models. These models were trained on the Common Objects in Context (COCO) dataset [9] and are presented in TFOD with their metrics in mAP and inference speed. In addition to the provided models, TFOD offers flexibility in the customization of both the processing and the training parameters. In the current study, an exploration was conducted regarding the development of a defect detection model, regarding the produced errors in antennas performed in the assembly process. For this reason, EfficientDet-Lite models were examined, since they achieve high metrics, with a small execution time.

The main contribution of this research work is the investigation of the modest EfficientDet-Lite versions and the application of the most capable for defect detection in the real time production process. In addition, the investigated models were further compared to the YOLO v5 that is a well-known detector used in various domains. In this current study, the considered defects in antennas are 1) metal cracks, 2) plastic cracks, and 3) housing imperfections. The proposed model provides remarkable results and can detect malfunctions effectively.

The rest of this paper is organized as follows. In section 2, the dataset, the methodological framework, and the models developed are presented. Section 3 presents the derived results. Section 4 includes the discussion and conclusion of the study.

2 MATERIALS AND METHODS

2.1 Dataset

The employed dataset in this work contains images directly acquired from (1) production and (2) laboratory measurements, using 2D area scan cameras. The cameras used in this study are a FLIR Blackfly S BFS-PGE-200S6C and a Baumer VCXG-241C. The output resolution of the sensors in width and height are 4401×2898 for the FLIR and 4096 × 3000 for the Baumer sensor, respectively. The dataset acquisition was focused on the defective samples that were scarce. Overall, the acquired dataset contains 161 defective samples and 25 healthy ones.

The defects were manually annotated with the use of LabelImg software to produce the required XML files in the PASCAL VOC [10] format. The defect labels were assigned by experts, and belong to three groups, as mentioned earlier: a) metal crack, b) plastic crack, and c) housing imperfection.

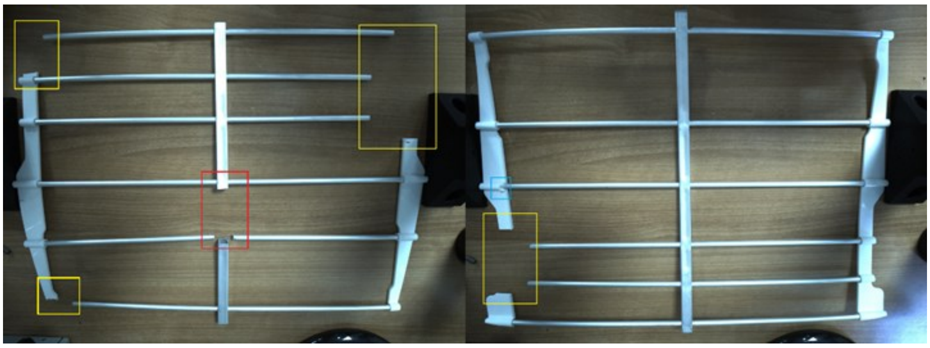

Figure 1: Our corresponding dataset contains three categories (a)metal crack (Red) (b) plastic crack (Yellow) (c) housing imperfection (Blue).

2.2 EfficientDet

The EfficientDet algorithm is a recent methodology for object detection. It can reach higher accuracy and efficiency, in comparison to other state-of-the-art object detectors. EfficientDet is actually a family of modes, that includes eight different variants. The accuracy and running time depend on model size [10] [11]. EfficientDet is known for outperforming benchmark models like CNNs (Convolutional Neural Networks) as it utilizes a small number of parameters [12]. EfficientDet uses EfficientNet as its backbone network, Bi-FPN as its neck, and convolutional layers as its head. Regarding the backbone network (EfficientNet), it can obtain reliable results. EfficientNet is proposed by Google, and it contains a scaling strategy, to balance network width and depth and image resolution. EfficientNet can perform surprisingly well and accomplish adequate accuracy and efficiency [12]. Regarding the neck, BiFPN is a weighted bi-directional feature pyramid network, which is equipped with learnable weights to extract the features from input images by employing top-down and bottom-up multi-scale fusion. EfficientDet contains a compound scaling method, that evenly scales the width, depth, and resolution for the feature network, the backbone network, and the box prediction simultaneously. The combination of EfficientNet as the backbone, Bi-FPN, and the compound scaling method produces a new category of object detectors that utilize fewer parameters and require, therefore, a smaller number of floating-point operations (FLOPs). More specifically, EfficientDet utilizes 28% fewer FLOPs than YOLOv3 and 30% fewer FLOPs than RetinaNet. According to the literature, the current object detectors are classified as single- or two-stage models, depending on whether they employ a region-of-interest protocol. Two stage-detectors are more accurate, while one-stage detectors are simpler and more lightweight. Recent trends show that one-stage detectors are preferred due to their efficiency and clarity. A difficulty that is often observed in object detection is the calculation and demonstration of multi-scale features. For this reason, former object detection algorithms predict with the support of the pyramidal feature hierarchy, which is extracted from backbone networks. An innovative methodology is feature pyramid network (FPN) that it introduces a top-down process for multi-scale features [13]. The EfficientDet-Lite models considered in this work, contain five versions [0-4] and belong to an object detection family of models derived from the EfficientDet architecture [14].

2.3 YOLOv5

You Look Only Once (YOLO) belongs to stage detector algorithms and is among the most popular object detection algorithms. Redmond et al. [2] proposed YOLO in 2016, which was an innovation for real-time object tracking at the time. Similar architectures to YOLO are quite complex since each component needs to be trained separately. For this reason, YOLO is based on regression methodology. YOLO's architecture contains 24 convolutional layers, with two fully connected layers. However, YOLO has experienced issues with generalization, especially when the objects are small in size, or the image has different dimensions.

YOLOv5 is a recent variant of the YOLO family that contains fewer parameters [15]. It consists of three main parts: a) the Backbone, b) the Neck, and c) the Head. Regarding the Backbone of YOLOv5, it utilizes Cross Stage Partial Networks (CSPNets) with DarkNet, which is known as the CSPDarknet. CSPDarknet has demonstrated an efficient improvement regarding processing speed. Regarding the Neck of YOLOv5, it uses Path Aggregation Network (PANet) as a parametric polymerization mechanism. In general, the neck of a network is applied for the generation of feature pyramids, for the generalization of object scaling, i.e., the detection of an object in different images with different sizes and backgrounds. In PANet, the feature grid is joined with the feature layers, where the information can be transmitted to the proposed subnetwork. Regarding the head of YOLOv5, it generates anchor boxes for the utilization of feature maps and extracts bounding boxes with class probabilities [16].

2.4 Methodology

The methodology employed in this work consists of the following steps:

1. **Data Collection**: The dataset utilized in this study consists of images directly acquired from production and laboratory measurements, obtained with 2D area scan cameras. In overall, the acquired dataset contains 161 defective samples and 25 healthy ones. Data were collected from a real industrial environment, provided by Televes.
2. **Image Annotation:** Expert knowledge was employed to manually annotate the images and identify all possible defects. Annotated images were needed to be used as input to the training process. The Python library "labelimg" [17] was used for the annotation of images. The location of each defect was marked with a bounding box.
3. **Splitting:** The dataset into training, testing and validation subsets. Following common practice, the training and validation datasets were employed for the development of the model, while the testing dataset was exploited for metric evaluation and performance assessment.
4. **Training:** Training the object detection model refers to the injection of the images along with the according annotated labels of each defect for the extraction of metrics. The training and validation datasets were employed to calculate the parameter values of the model.
5. **Evaluation:** The testing dataset, unseen so far by the model, was utilized to calculate specific metrics and assess the model's efficiency and stability.

3 RESULTS

This study attempts to address an automatic pipeline for detecting defects. For this purpose, many experiments were conducted to the optimal configuration for the provided dataset, while preserving generality. The developed model was on a workstation with the following technical specifications: AMD Ryzen 7 5800H CPU with Radeon Graphics, NVIDIA Ge-Force RTX 3060 GPU and 16 GB RAM. The model was developed in Python 3.9.13, with TensorFlow 2.9.3, Keras 2.9.0, and tflite_model_maker 0.3.4. A thorough exploration was performed by testing different values for batch size (4, 8, 16, 32) and epochs like (400, 500, 600, 700, 800, and 900) for different versions of EfficientDet-Lite models.

Common, yet robust, metrics were employed to assess performance of the model:

1. **Precision.** This metric demonstrates the ratio of the instances that are predicted as positives and are truly positive. The average precision for every category of objects is defined as AP.
2. **Intersection over Union (IOU)**. This indicates the deviation of the predicted result from the actual output. It can be calculated by the intersection of the bounding box and the true annotation box over the union of two boxes [18].

The best-performing models are listed in Table 1, along with the values of the performance metrics.

Table 1: Performance metrics of the various EfficientDet-Lite models and YOLOv5. (class1: metal crack, class2: plastic crack, class3: housing imperfection)

Model architecture	AP	AP50	AP75	AP of class1	AP of class2	AP of class3	Training Time
Lite0	20.48	50.38	11.68	12.3	23.15	18.35	4955 sec
Lite1	22.59	58.35	25.15	11.45	30.56	25.12	7025 sec
Lite2	19.71	73.68	12.35	12.20	21.44	18.49	6044 sec
YOLOv5	69.10	74.20	43.20	23.40	36.50	69.60	6052 sec

It is clear that EfficientDet-Lite2 has outperformed the rest of the models. The AP50 of 73.68% indicates that the proposed model can correctly detect the defects. The proposed model proved its ability to differentiate between the three possible outputs and correctly detect the defects. It produced satisfactory results, regarding the small dataset. Results in Figure 2 support that the proposed model correctly detects the plastic crack with adequate precision, even though the background is unclear, very complex and with poor lighting conditions, which could easily lead to misinterpretations.

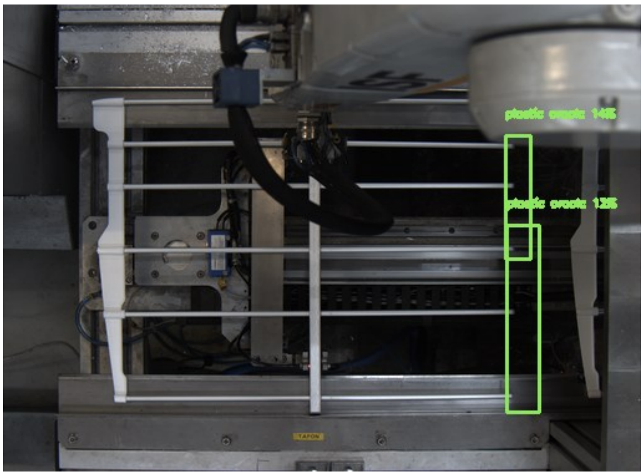

Figure 2: Efficient defect detection using EfficientDet-Lite2 on an image with complex background and poor lighting conditions. The model successfully detects the missing part on the right side.

In Figure 3, it is demonstrated that the model detected a plastic crack (broken part) with 21% precision at the bottom right corner. Both predictions were made in real time.

Figure 3: Efficient crack detection (broken part) using EfficientDet-Lite2.

4 SUMMARY

Efficient defect detection in real-time production can benefit from artificial intelligence methodologies for the different stages of the manufacturing process. Artificial intelligence technologies can be adjusted to production conditions and towards Zero Defect Manufacturing. In smart manufacturing, the most popular methodologies are based on deep learning classification, image analysis and object detection.

This paper developed and assessed an automatic real-time defect detection system based on a deep learning methodology. EfficientDet-lite variants were utilized with different parameters regarding batch size and epochs. The best results were obtained with EfficientDet-Lite2 with batch size 4 and 700 training epochs. The main advantage of EfficientDetLite-2 is that it can detect all classes correctly, even in images with a complex background and poor lighting conditions. The performance of the rest of the EfficientDet-Lite variants were not significantly inferior regarding the average precision. The performance of the proposed model is validated with various IoU levels of precision and with higher extracted average precision for each possible class. The proposed method provides efficient results to defect detection for a complicated dataset, coming from a real industrial environment, that has not been explored before. The performance of YOLOv5 was satisfactory regarding training time and parameters but exhibited inferior defect detection capabilities.

Future extensions of this work will focus on experimental analysis of other state-of-the-art models, including the rest of YOLO variants.

5 ACKNOWLEDGEMENTS

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

REFERENCES

[1] D. Ben-Arieh, R. Ranjan Kumar, and M. K. Tiwari, "Analysis of assembly operations' difficulty using enhanced expert high-level colored fuzzy Petri net model," *Robot. Comput.-Integr. Manuf.*, vol. 20, no. 5, pp. 385–403, Oct. 2004, doi: 10.1016/j.rcim.2004.03.002.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[3] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo algorithm developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2022.

[4] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection." arXiv, Jul. 27, 2020. Accessed: Dec. 28, 2022. [Online]. Available: http://arxiv.org/abs/1911.09070

[5] 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

[6] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv Prepr. ArXiv201011929*, 2020.

[7] T. Tziolas *et al.*, "Wafer Map Defect Pattern Recognition using Imbalanced Datasets," in *2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 2022, pp. 1–8. doi: 10.1109/IISA56318.2022.9904402.

[8] 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.

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

[10] "EfficientDet for fabric defect detection based on edge computing - Shaojun Song, Junfeng Jing, Yanqing Huang, Mingyang Shi, 2021." https://journals.sagepub.com/doi/full/10.1177/15589250211008346 (accessed Feb. 08, 2023).

[11] A. M. Nagy and L. Czúni, "Classification and Fast Few-Shot Learning of Steel Surface Defects with Randomized Network," *Appl. Sci.*, vol. 12, no. 8, Art. no. 8, Jan. 2022, doi: 10.3390/app12083967.

[12] M. N. Ab Wahab, A. Nazir, A. T. Zhen Ren, M. H. Mohd Noor, M. F. Akbar, and A. S. A. Mohamed, "Efficientnet-Lite and Hybrid CNN-KNN Implementation for Facial Expression Recognition on Raspberry Pi," *IEEE Access*, vol. 9, pp. 134065–134080, 2021, doi: 10.1109/ACCESS.2021.3113337.

[13] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection." arXiv, Jul. 27, 2020. doi: 10.48550/arXiv.1911.09070.

[14] "Object Detection with TensorFlow Lite Model Maker," *TensorFlow*. https://www.tensorflow.org/lite/models/modify/model_maker/object_detection (accessed Feb. 19, 2023).

[15] M. L. Mekhalfi, C. Nicolò, Y. Bazi, M. M. A. Rahhal, N. A. Alsharif, and E. A. Maghayreh, "Contrasting YOLOv5, Transformer, and EfficientDet Detectors for Crop Circle Detection in Desert," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2021.3085139.

[16] "Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model :: BioResources." https://bioresources.cnr.ncsu.edu/ (accessed Feb. 19, 2023).

[17] "labelImg: LabelImg is a graphical image annotation tool and label object bounding boxes in images." Accessed: Feb. 08, 2023. [Online]. Available: https://github.com/tzutalin/labelImg

[18] J. Wang, "Lightweight and real-time object detection model on edge devices with model quantization.," *J. Phys. Conf. Ser.*, vol. 1748, no. 3, p. 032055, Jan. 2021, doi: 10.1088/1742-6596/1748/3/032055.