2023 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Tomar, Portugal – April 26-27, 2023

2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) | 979-8-3503-0121-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICARSC58346.2023.10129634

Quality Control of Casting Aluminum Parts: A Comparison of Deep Learning Models for Filings Detection

Rui Nascimento*†, Tony Ferreira*, Cláudia Rocha*, Vítor Filipe*†, Manuel F. Silva*‡,
Germano Veiga*, Luis Rocha*
*INESC TEC - Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência,
Porto, Portugal
†School of Science and Technology,Universidade de Trás-os-Montes e Alto Douro,
Vila Real, Portugal
‡Instituto Superior de Engenharia do Porto,Instituto Politécnico do Porto,
Porto, Portugal
0000-0001-7410-517X, 0000-0002-6688-406X, 0000-0001-7254-0346, 0000-0002-3747-6577, 0000-0002-0593-2865,
0000-0003-3044-6938, 0000-0002-8680-4290

***Abstract*—Quality control inspection systems are crucial and a key factor in maintaining and ensuring the integrity of any product. The quality inspection task is a repetitive task, when performed by operators only, it can be slow and susceptible to failures due to the lack of attention and fatigue. This work focuses on the inspection of parts made of high-pressure die-cast aluminum for components of the automotive industry. In the present case study, last year, 18240 parts needed to be reinspected, requiring approximately 96 hours, a time that could be spent on other tasks. This article performs a comparison of four deep learning models: Faster R-CNN, RetinaNet, YOLOv7, and YOLOv7-tiny, to find out which one is more suited to perform the quality inspection task of detecting metal filings on casting aluminum parts. As for this use-case the prototype must be highly intolerant to False Negatives, that is, the part being defective and passing undetected, Faster R-CNN was considered the best-performing model based on a Recall value of 96.00%.**

***Index Terms*—Quality control, Convolutional Neural Networks, Filings detection, Casting aluminum, Automotive industry**

I. Introduction

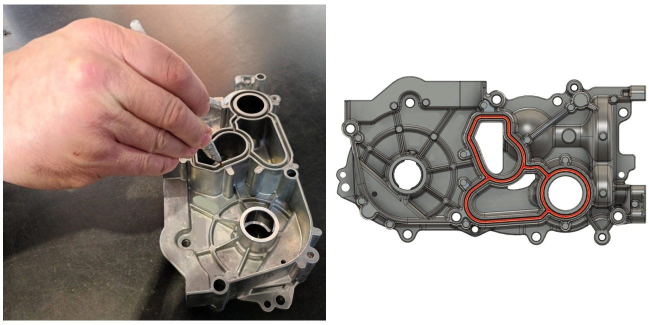
The automotive industry is inserted in an increasingly demanding and competitive market. For that reason, they seek to integrate new techniques into their production lines that allow them to differentiate themselves from their competitors. These techniques allow an increase in the productivity and quality of their products, which contributes to the reduction in customer complaints. Product quality is a very important factor, and more and more consumers are taking this factor into account, which is why it is important that existing inspection methods in production lines are able to respond to the demand imposed by consumers [1]–[3]. Quality control is a repetitive process typically carried out by operators that makes the process slow and susceptible to failures due to the lack of attention and fatigue, which end up influencing the entire process and causing the advance of non-conforming parts in the company's production process. Machine vision is one of the techniques that can be used for quality inspection, allowing a quick inspection of parts and the detection of defects. The machine vision inspection can be faster and provide supplementary assistance to the factory floor operator, improving speed and decreasing failures.

In the present case, the high-pressure die-cast aluminum parts are manufactured by an injection molding process followed by a machining phase to ensure accurate dimensioning. Currently, after the machining process, the part is subjected to a tightness control operation. As soon as this operation is finished, the operator takes the part and, manually, checks and removes, with the help of a tool, any detachable particle from inside the entire slot, as shown in Fig. 1a. In the proposed case study, machine vision is used to inspect the quality of cast aluminum automotive parts, detecting existing filings in the tear area, highlighted in red in Fig. 1b. As this is a structural area where another part will later be fitted, it is very important that the slot is completely unobstructed to ensure the correct fitting of the parts.

In order to avoid delivery of non-conforming parts to the customer, each time a non-conforming part is detected on the production line, the entire batch is suspended, forcing a re-inspection of the same. Regarding the model of the part used in the case study, 24 non-compliant parts were detected last year. Each of the non-conforming parts detected leads to the suspension of approximately 760 parts and all of them are re-inspected. On average, the operator spends 4 hours re-inspecting the 760 parts; therefore, in total, last year, 18240 parts were reinspected, and approximately 96 hours were spent re-inspecting parts, that could be used by the operator to perform other tasks.

The work presented in this paper focuses on the development of an automatic quality inspection system for the

2023 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Tomar, Portugal – April 26-27, 2023



(a) (b)

Fig. 1: Cast aluminum automotive part: (a) quality inspection process manually performed by an operator, and (b) 3D model with a tear zone highlighted in red.

detection of filings in the tear zone based on object detection, in order to eliminate non-conforming parts sent to the customer and the number of hours spent on re-inspecting them. A performance comparison was made between four different models, Faster R-CNN [4], RetinaNet [5], YOLOv7, and YOLOv7-tiny [6] for detecting filings in casting aluminum parts.

Section II presents the state of the art regarding inspection systems using Convolutional Neural Networks (CNN) for the automotive industry and metal parts, while section III describes the hardware and the deep learning models used in the inspection system. Section IV provides an analysis of a set of experimental results and Section V summarizes the main conclusions and future work.

II. State Of The Art

The detection and classification of surface defects differ from case to case, usually requiring custom-made solutions to ensure that the algorithms used learn the visual discriminative features representing the data. In recent years, there has been a lot of progress in CNN-based approaches, which has led to the proposal of various target-detection algorithms. These can be classified into single-stage detection algorithms, such as the You Only Look Once (YOLO) series and single-shot multibox detector (SSD), or two-stage detection algorithms, including Region-CNN(R-CNN), Fast Region-based CNN (Fast R-CNN), Faster Region-based CNN (Faster R-CNN), among others. The use of CNN for quality inspection in automotive industries and in metal parts is an active field of research with many different approaches.

Wang *et al.* [7] proposed an 11-layer CNN for defect detection that can automatically extract powerful features with less prior knowledge about the images, and is robust to noise. The model is trained and validated using the DAGM dataset, which contains image samples from six classes that differ in terms of background texture, achieving an overall detection accuracy of 99.8%. Fu *et al.* [8] proposed a method that uses the pre-trained SqueezeNet as a backbone for steel surface defect recognition, requiring only a small amount of defect-specific training samples to achieve accurate defect recognition. The training and validation of the proposed model are done using the NEU steel surface benchmark dataset, and the results show significantly higher recognition accuracy compared with the state-of-the-art steel surface defect classifiers. Xu *et al.* [9] proposed an intelligent recognition model for surface defects in a copper strip. The dataset used in their work contains 2400 images of surface defects, each defect is classified as a single or line mark, black spot, concave-convex pit, edge crack, hole, insect spot, peeling, or smudge. Four CNN models were adopted, with EfficientNet having the best overall performance with a recognition accuracy rate of 93.05%.

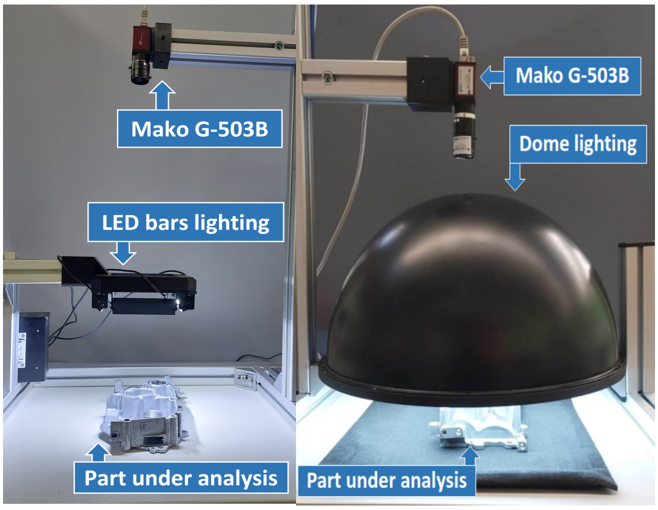
In the inspection of metal parts, Block *et al.* [10] proposed a framework based on RetinaNet. A custom dataset containing 31 videos and 31504 images, with defects classified as mild or severe, was used in the training and validation, achieving a mean Average Precision (mAP) of 76.21% to detect and classify mild and severe defects. Sun *et al.* [11] proposed a model based on adaptive multiscale image collection (AMI(c), using VGG-16 for the inspection. The custom dataset used in training and validation has a total of 1274 images divided into 4 classes: normal, indentation, scratch, and pitted surface. The average inspection precision of the proposed algorithm was 98.97%.

For the inspection of aluminum parts, Wei *et al.* [12] proposed a multiscale defect-detection network based on the Faster R-CNN. A custom dataset with a total of 3005 images was used to train and validate the model. It contained ten types of defects: non-conducting, scratch, corner leak, orange peel, leakage, jet, paint bubble, crater, parti-color, and dirty point. The results show an mAP of 75.8%. Du *et al.* [13] proposed a defect detection system based on Faster R-CNN. For the model training and validation, 2236 X-ray images of defective automobile parts were collected. The results show a 40.9% improvement in the mAP. Mery [14]evaluated eight state-of-the-art deep object detection models, based on YOLO, RetinaNet, and EfficientDet, proposing a training strategy that used a low number of defect-free X-ray images. For the experiments on the GDXray dataset, series C0001, and YOLOv5-methods were used. YOLOv5s was the model which obtained the best results, with an average precision of 90% and an F1 factor of 91%. Wang *et al.* [15] proposed a model based on YOLOv5. The Ali Tianchi dataset was used, containing 3098 images of seven types of defects (concavity, dirty spot, orange peel, non-conducting, scrape, under-screen, and embossing) for training and testing, achieving 87.4% detection.

To the best of the author's knowledge, there are still no solutions for quality inspection in casting aluminum parts, in particular, the detection of filings. So, it is not possible to compare different types of CNN to verify which one is the most suitable for the detection of filings.

III. Methodology

In this paper, four deep learning models were used to detect the presence of filings in casting aluminum parts. Its performances were then compared to find out which one is



(a) (b)

Fig. 2: Laboratorial prototype setup for image acquisition with different lightning types: (a) LED bars with adjustable angular bars; (b) dome light.

best suited to the task. This section is organized as follows. A description of the hardware used in this work is given in Section III-A, and in Section III-B is described the custom dataset. Section III-C gives a theoretical background of the models used in this work and explains how they are evaluated.

A. Image Acquisition and Processing

For the development of this work, a computer with Ubuntu 20.04 operating system, a 12th Generation Intel Core i7 CPU @3.5Ghz processor, an NVIDIA RTX3060 graphics card with 6GB of running memory was used, and CUDA 11.7 has been installed to speed up the computation. A Mako G-503B, equipped with a 12 mm focal length lens, was used to acquire images. The laboratory setup prototype includes two types of lightning, LED bars, and dome lightning, shown in Fig. 2a and Fig. 2b, respectively. This was tested since the parts under analysis are highly reflective, and proper lighting is necessary to reduce the reflections and shadows to the minimum possible. Among the lighting tested, dome lighting was the chosen one, as it showed the best results.

B. Dataset

For this work, it was necessary to create a customized dataset since there are no available datasets with filings as defects. These filings vary in length, shape, and reflective characteristics, as shown in Fig.3. The dataset comprises 500, images with a resolution of 2592x1944 pixels, manually labeled, using LabelImg [17]. Fig. 4 shows some examples of images of the dataset. Since the number of non-conforming parts obtained was small, there was a need to simulate them, by manually placing the filings in the tear zone.

In order to make the dataset more robust, images were also acquired with different orientations. The dataset was split, randomly, into three parts: training, validation, and testing, which translates to 80%, 10%, and 10% of the dataset, respectively.

Fig. 3: Example of the shape and size of filings to be detected.

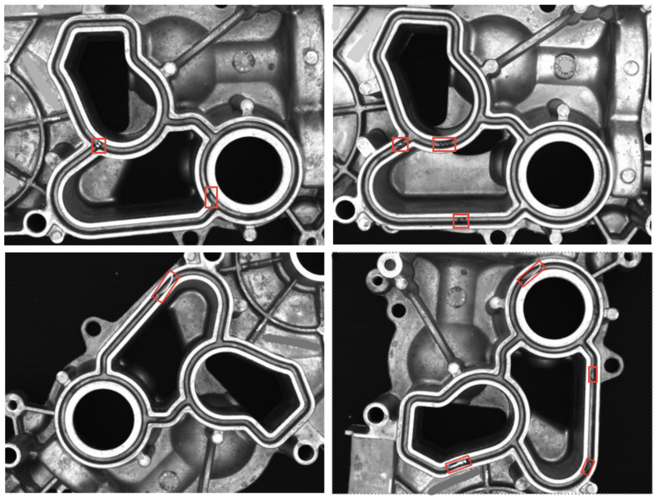


Fig. 4: Example of dataset images.

C. Model Implementation

To detect filings on aluminum casting parts, four CNN models were chosen based on the results presented in Section II: three one-stage object detectors, RetinaNet, YOLOv7, and YOLOv7-tiny, and one two-stage object detector, Faster R-CNN. For the training of the models, the Python programming language with the aid of the Pytorch library was used. RetinaNet and Faster R-CNN also resort to Detectron2 library [16].

RetinaNet [5] is a combination of networks consisting of a backbone and two task-specific subnetworks. The architecture of RetinaNet can be broken down into three blocks:

1) Backbone Network - responsible for computing convolutional feature maps over the entire image. Feature Pyramid Network (FPN) is implemented as the backbone network, providing a rich, multi-scale feature pyramid by implementing a top-down approach with lateral connections;
2) Sub-network for Object Classification - is a Fully Convolutional Network (FCN) attached to each FPN level for object classification;
3) Sub-network for Object Regression - is attached to each feature of the FPN in parallel to the classification

subnetwork.

Faster R-CNN [4] is based on the R-CNN family. These networks usually consist of four layers: the Region Proposal Algorithm that generates the bounding boxes or locations of possibles objects in the image, the Feature Generation Stage to obtain features of the objects found, the Classification Layer responsible for predicting which class the found object belongs to, and Regression Layer that makes the coordinates of the object bounding box more precise. Compared to the others of the R-CNN family, this model uses another convolutional network, Region Proposal Network (RPN), to generate the regions proposals, causing an overall improvement in feature representation.

YOLOv7 belongs to the YOLO family. Compared to the other YOLO models, two major changes were made in its architecture and at the trainable bag-of-freebies level, which refers to improving the models accuracy without increasing the training cost [6]. On the architecture level, YOLOv7 reformed its architecture by integrating the Extended Efficient Layer Aggregation Network (E-ELAN), allowing the model to learn more diverse features for better learning. The architecture of the models it is derived from was also concatenated to allow the model to meet the needs of different inference speeds. YOLOv7-tiny is a basic model optimized for edge computing. Compared to the other versions, YOLOv7-tiny uses leaky ReLU as the activation function, while other models use SiLU as the activation function.

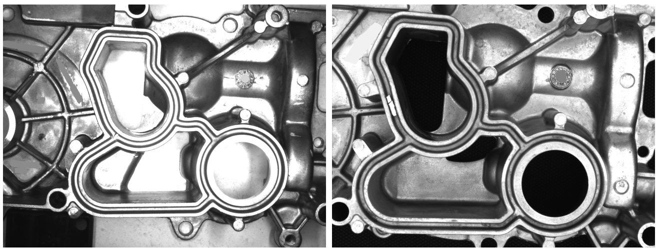
In order to avoid over and under-fitting the models, TensorBoard was used to analyze in real-time the data during the training. Once the loss curve as been seen as stable, the training was set as complete and the best model was used. Once the training process is complete, the weights are used to evaluate the performance of each model in terms of its precision, recall, and mAP, for different confidence thresholds, F1-score, and inference time of each model. For the calculation of the results, a False Positive (FP) was considered as wrongly detecting a filing and False Negative (FN) not detecting a filing. As these values of precision, recall, F1-score, and mAP are computed according to the True Positives (TP), FP, and FN, it is worth mentioning that the Intersection over Union (IoU) threshold defined for this classification corresponds to 50%.

IV. Results

This section is organized as follows. Firstly, the results of the type of light comparison tests conducted in the laboratory setup prototype are presented in Section IV-A, and then a comparison of the analysed models' performance is carried out in Section IV-B.

A. Lighting analysis

For the assembly of the laboratory setup, two types of lighting were tested: LED bars and dome lights. These were chosen due to their characteristics, which allowed acquiring an image of the part with the least possible reflections and shadows. LED lighting with angular bars that can be adjusted allows the angle of light incidence on the part to be changed, and dome lighting allows illuminating the part uniformly. The result of the parts lighting, for each type of lighting, is visible in Fig. 5. When comparing the two images' quality, dome lighting performs best at minimizing reflections and shadows; for this reason, it was the lightning chosen.



(a) (b)

Fig. 5: Lighting types' results: (a) LED bars with adjustable angular bars; (b) dome light.

B. Performance evaluation

As previously stated, four different models were trained: Faster R-CNN, RetinaNet, YOLOv7, chosen based on the State of the Art analysis, and YOLOv7-tiny, a lighter version of the YOLOv7 model, chosen due to hardware limitations. Faster R-CNN and RetinaNet use Resnet-101-FPN as the backbone.

Firstly, the trade-off between the input image resolution and the batch size was analyzed, translating to faster feature extraction for better image quality. GPU memory was used as a constraint to define the minimum resolution with the maximum batch size, and vice versa, to see which version provided better results. Therefore, the input resolution was successively decreased, maintaining the aspect ratio of the original 2592x1944 pixels image, until the training could be completed at the maximum capacity of the GPU. By resizing, the image it was possible to conclude that, image size significantly impacts the performance, since the defects to be detected ranged from 1 mm to 12 mm, approximately. For that reason, the images were resized to 1600x1200 pixels.

Compared to the other models, YOLOv7-tiny is a lighter model optimized for GPU, which allowed it to be trained with a larger batch size than the others, for the same image size. All models were trained at a learning rate of 0.0025 with Adam optimizer. The data augmentation performed was the same for the four models, horizontal and vertical flips, and brightness variation.

By using Detectron2 to train the Faster R-CNN and RetinaNet models, the number of iterations to train the model was defined instead of specifying the number of epochs, as in the YOLOv7 and YOLOv7-tiny. An iteration corresponds to a training cycle, one forward and one backward pass, with the number of images equal to the batch size. An epoch implies the use of the whole dataset for training. In this case, 400 training images were used and a batch size of 1 for Faster R-CNN, RetinaNet, and YOLOv7, and a batch size of 5 for YOLOv7-tiny, due to GPU, due to GPU constrain memory. Analyzing the box loss curves for training and validation, presented in Fig. 6 and Fig. 7, respectively, it is possible to verify that for the Faster R-CNN and RetinaNet models the curves stabilized between 18000 and 19000 iterations, which is equivalent to 45 epochs, and for the YOLOv7 and YOLOv7-tiny models the curves stabilized around 180 and 80 epochs, respectively. As previously mentioned, during the training, TensorBoard was used to monitor the model training. Once the loss curves were stable, the training has been stopped and the best model was used.

Table I shows the comparative results between the four models in the test dataset, which corresponds to 50 images. Regarding inference time, Faster R-CNN and YOLOv7-tiny were, respectively, the slowest (171 ms) and the quickest (12.2 ms) to make a prediction. Even then, all of them are fast enough for the requirements of this use case, such that inference time ended up not being a rejection criterion. By comparing the results, it is possible to conclude that, the Faster R-CNN model obtained the best precision and recall, 96.96% and 96%, respectively. This means that the model was the best at correctly classifying and detecting the filings. Because of that the F1-score is also higher for Faster R-CNN, 96.00%. Analyzing the mAP values of the four models it is possible to conclude that YOLOv7 achieves the best results 97.70%, and 59.20%, for mAP@0.5 and mAP@0.5:0.95, respectively. This means the YOLOv7 model is more stable and consistent across different confidence thresholds. In this case study FN are more critical than the FP, and taking into account that the mAP difference between the models is not very large, the model that obtained the highest recall, Faster R-CNN, was considered more adequate.

The dataset split of the test batch was used to perform the inference with the trained models, to see their performances in detecting the filings. In Fig. 8 is shown the bounding boxes of each filing detected in red, with the corresponding classification and confidence value, on top of each box. Fig. 8(a) shows, in green, the original labeled images being considered the ground truth of the experiments. By comparing the inference results is possible to conclude that, as a result of having a higher recall, the confidence of the filing prediction is higher in Faster R-CNN and RetinaNet, as shown in Fig. 8(b) and Fig. 8(c), respectively. However, the precision of locating the filing is similar, yet lower than YOLOv7 and YOLOv7-tiny, presented in Fig. 8(d) and Fig. 8(e), respectively.

V. Conclusions and Future Work

Product quality is an increasingly important factor taken into account by consumers in the automotive industry. It is required that the inspection methods on production lines are able to respond to the demand imposed by consumers. As no previous work was found where the detection of filings in casting aluminum parts was carried out, four deep learning-based models were trained to detect the filings, and their performance was compared. Given that it is unacceptable for the system to classify a part as defect-free when it contains defects, special attention was given to the recall metric, on which the Faster R-CNN model achieved the higher percentage: 96.00%. Despite the mAP being lower for Faster R-CNN (56.84%), the four models had similar mAP values. In the inference results, it was possible to conclude that Faster R-CNN and RetinaNet obtain higher values of confidence in classifying the defect, and the four models presented similar precision values. The results also show that Faster R-CNN and YOLOv7-tiny were the slowest (171 ms) and the quickest (12.2 ms), respectively. However, all of them are fast enough for the requirements of this case study. For that reason, Faster R-CNN was considered to be most suitable for this case study. A comparison between the two types of lighting was also made, to find out which one was more adequate for the inspection of these cast aluminum parts. The lights chosen presented the characteristics necessary to illuminate the parts with the minimum possible reflections and shadows. By comparing the acquired images, it was possible to conclude that dome lighting presented the best results, illuminating the part evenly and reducing the reflections and shadows.

For future work, the chosen model will be used for the continuation of the project, where more training can be done to improve its performance and detect other common defects in this type of part, such as scratches, dents, and cracks, among others, and increase the hardware processing capability to achieve better performance results. CNN-based semantic segmentation networks can be used to obtain a detailed characterization of the tear zone, and could be considered to perform a comparison with CNN-based object detectors. Moreover, a collaborative robotic manipulator will be integrated to automate the process of exchanging the parts.

Acknowledgment

This work is co-financed by Component 5 - Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021 - 2026, within project Hi1_reV, with reference 64.

References

[1] C.-C. Huang and X.-P. Lin, “Study on Machine Learning Based Intelligent Defect Detection System”, MATEC Web of Conferences, vol. 201, p. 01010, 2018.

[2] A. Kazemian, X. Yuan, O. Davtalab, and B. Khoshnevis, “Computer vision for real-time extrusion quality monitoring and control in robotic construction”, Automation in Construction, vol. 101, pp. 92–98, May 2019.

[3] S. S. Martínez, J. G. Ortega, J. G. García, and A. S. García, “A machine vision system for defect characterization on transparent parts with non-plane surfaces”, Machine Vision and Applications, vol. 23, no. 1, pp. 1–13, July 2010.

[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. arXiv, 2015.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection”. arXiv, 2017.

[6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. arXiv, 2022.

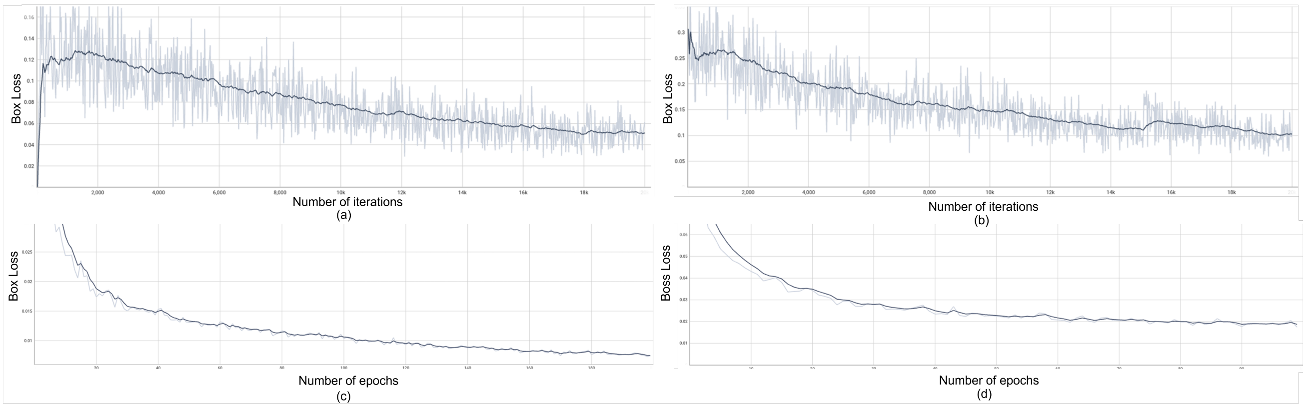


Fig. 6: Training Box Loss curves: (a) Faster R-CNN, (b) RetinaNet, (c) YOLOv7, and (d) YOLOv7-tiny.

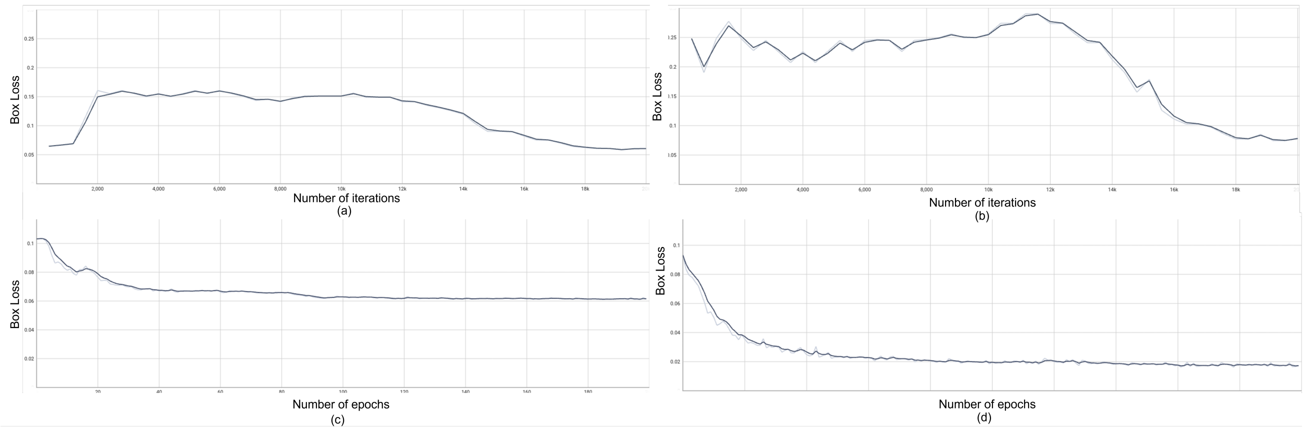


Fig. 7: Validation Box Loss curves: (a) Faster R-CNN, (b) RetinaNet, (c) YOLOv7, and (d) YOLOv7-tiny.

TABLE I: Comparison results between the four proposed models

Model	Batch size	Precision (%)	Recall (%)	F1-score (%)	mAP@0.5(%)	mAP@0.5:0.95(%)	GPU memory* (MiB)	Inference time (ms)
Faster R-CNN	1	95.09	97.00	96.04	96.65	55.76	5637	171.00
RetinaNet	1	96.87	93.00	94.90	92.97	49.86	4513	165.00
YOLOv7	1	96.90	93.00	94.91	97.70	59.20	5670	50.30
YOLOv7-tiny	5	94.00	94.00	94.00	95.80	55.90	5429	12.20

* GPU memory required during model training, taking into account the considered batch size.

[7] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control", The International Journal of Advanced Manufacturing Technology, vol. 94, no. 9–12, pp. 3465–3471, August 2017.

[8] G. Fu *et al.*, "A deep-learning-based approach for fast and robust steel surface defects classification", Optics and Lasers in Engineering, vol. 121, pp. 397–405, 2019.

[9] Y. Xu, D. Wang, B. Duan, H. Yu, and H. Liu, "Copper Strip Surface Defect Detection Model Based on Deep Convolutional Neural Network", Applied Sciences, vol. 11, no. 19, p. 8945, Sep. 2021.

[10] S. B. Block, R. D. da Silva, L. B. Dorini, and R. Minetto, "Inspection of Imprint Defects in Stamped Metal Surfaces Using Deep Learning and Tracking", IEEE Transactions on Industrial Electronics, vol. 68, no. 5, pp. 4498–4507, 2021.

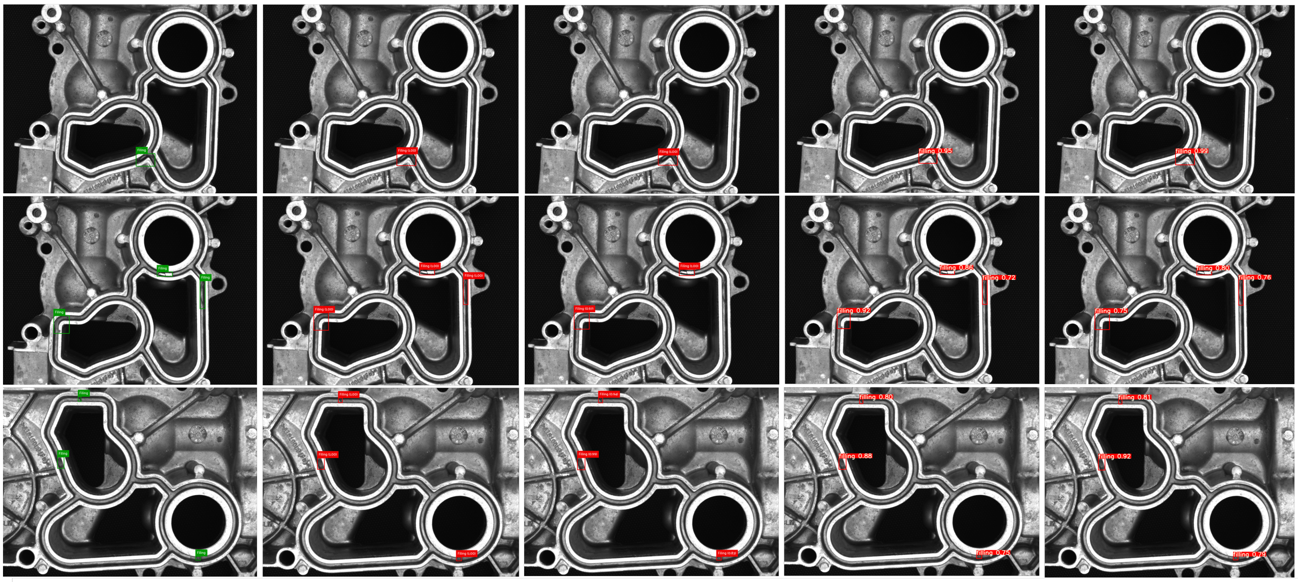
[11] J. Sun, P. Wang, Y.-K. Luo, and W. Li, "Surface Defects Detection Based on Adaptive Multiscale Image Collection and Convolutional Neural Networks", IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 12, pp. 4787–4797, 2019.

[12] R. Wei and Y. Bi, "Research on Recognition Technology of Aluminum Profile Surface Defects Based on Deep Learning", Materials, vol. 12, no. 10, p. 1681, May 2019.

[13] W. Du, H. Shen, J. Fu, G. Zhang, and Q. He, "Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning", NDT & International, vol. 107, p. 102144, October 2019.

[14] D. Mery, 'Aluminum Casting Inspection using Deep Object Detection Methods and Simulated Ellipsoidal Defects', Machine Vision and Applications, vol. 32, 05 2021.

[15] T. Wang, J. Su, C. Xu, and Y. Zhang, "An Intelligent Method for Detecting Surface Defects in Aluminium Profiles Based on the Improved



(a) (b) (c) (d) (e)

Fig. 8: Inference results: (a) Ground Truth, (b) Faster R-CNN, (c) RetinaNet, (d) YOLOv7, and (e) YOLOv7-tiny.

YOLOv5 Algorithm", Electronics, vol. 11, no. 15, 2022.

[16] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2", 2019. [Online]. Available: https://github.com/facebookresearch/detectron2.

[17] Tzutalin, 'LabelImg', 2015. [Online]. Available: https://github.com/heartexlabs/labelImg.