Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Analysis Detection of Real-Time Metallic Surface Defect Using MobileNetv2 and YOLOv3 on Raspberry Pi

Muhammad Noorazlan Shah Zainudin[1,*], Muhammad Idzdihar Idris[1], Soh Jun Kang[1], Tan An Chee[1], Teoh Yu Xian[1], Jamil Abedalrahim Jamil Alsayaydeh[1], Sufri Muhammad[2], Md Saifullah Razali[3]

[1] Centre for Telecommunication Research & Innovation, Faculty of Electronic and Computer Engineering and Technology, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia
[2] Faculty of Computer Science and Information System, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[3] PSB Academy, 6 Raffles Blvd, #03 - 200, 039594, Singapore

ABSTRACT

This work presents an innovative solution utilizing a Raspberry Pi detection system to identify any defects on metallic surfaces in real-time. Manual inspection has several limitations, including time-consuming, subjective assessments, and a higher probability of human error could compromise product quality, lead to potential failures, and result in substantial costs for manufacturers. The primary focus of this endeavour is to enhance manufacturing efficiency and reduce labour expenses by automating the defect identification process. This objective is realized by employing the YOLOv3-tiny and MobileNetv2 algorithms which are subsequently deployed on a Raspberry Pi to enable precise and swift defect detection on metallic surfaces. The implementation process involves training and testing the models on a computer, followed by their deployment onto the Raspberry Pi. Upon proper setup, the trained models are employed for real-time inferences, effectively identifying defects. Notably, while the MobileNetv2 exhibits impressive accuracy in classifying defect types above 0.9, it is found to be less efficient for real-time detection on the Raspberry Pi. In contrast, the YOLO model proves to be well-suited for real-time detection on this platform with above probability of 0.8 for selected types of defects. The successful integration of this model significantly transforms quality control and inspection procedures across various industries.

Keywords:
YOLOv3; MobileNetv2; Raspberry Pi; defect detection

1. Introduction

Industries heavily reliant on metallic surfaces, such as manufacturing, automotive, aerospace, and construction, place paramount importance on maintaining high-quality standards. Adhering rigorously to these standards is essential to ensure products align with required specifications and perform impeccably. The vulnerability of metallic surfaces to diverse defects and imperfections underscores the need for diligent quality control [1]. These flaws, ranging from punched holes and oil spots to crescent gaps and rolled pits, can significantly compromise the strength, durability, and

* *Corresponding author.*
E-mail address: noorazlan@utem.edu.my

https://doi.org/10.37934/araset.57.2.113

overall functionality of end products. Swift and accurate identification and resolution of such defects are imperative to uphold product excellence, prevent costly rework, and avert any potential customer discontentment. Overcoming these obstacles necessitates the establishment of robust quality control strategies. This entails harnessing sophisticated computer vision methods and employing a pre-trained machine learning model [2]. Through the utilization of these advancements, sectors can adeptly identify and categorize flaws present on metallic surfaces, facilitating swift rectification. Embracing this proactive methodology substantially diminishes the risk of supplying defective or below-par goods to clientele. As a result, customer contentment is amplified, instances of product recalls are curtailed, and the industry's standing and brand credibility remain intact.

The manufacturing factory serves as the operational hub for metal production. This realm encompasses a diverse array of processes and activities integral to crafting metal components and items. Within this domain, activities like metal cutting, machining, forming, welding, assembly, and finishing converge [3]. Here, adept artisans and technicians leverage specialized tools, machinery, and methodologies to metamorphose raw metal materials into precise and functional elements. This intricate process involves the use of equipment such as lathes, milling machines, computer numerical control (CNC) systems, presses, and welding stations to execute cutting, drilling, shaping, and fusion procedures. Furthermore, the pivotal roles of quality control and inspection are evident in upholding the stipulated standards and specifications for the manufactured metal goods. The manufacturing metal arena amalgamates technical prowess, artistry, and cutting-edge technologies to yield an extensive spectrum of metal components indispensable across industries such as automotive, aerospace, construction, and beyond. The objective of performing defect detection on metal is to detect and rectify any flaws or irregularities that could potentially undermine the quality, structural robustness, or operational effectiveness of metal elements. Imperfections can emerge at different junctures within the metal production journey, spanning activities like casting, forging, machining, welding, and fabrication [4].

Yet, defects detection on metallic surfaces presents considerable difficulties across diverse industries as prevalent techniques predominantly hinge on manual inspection. Nevertheless, manual inspection is fraught with several constraints, encompassing protracted procedures, subjectivity in evaluations, and elevated risk of human inaccuracies. These constraints have the potential to undermine product excellence, trigger potential malfunctions, and impose significant financial burdens on manufacturers. The practice of manually inspecting metallic surface defects remains widespread in numerous industries, impeding the overall effectiveness and dependability of the inspection workflow. Relying on manual inspection techniques results in prolonged inspection duration, labour-intensive processes, and an escalated potential for inaccuracies. These hurdles impede industries' capacity to ensure uniform product quality, adhere to regulatory norms, and furnish customers with flawless goods. As a result, there exists a need for an automated and resilient defect detection system capable of surmounting these constraints and delivering precise, on-the-spot evaluations of metallic surface states. Hence, an automated detection of defects on metallic surfaces mandates the creation and fusion of cutting-edge computer vision methodologies and machine learning algorithms [5]. The system must possess the capacity to adeptly manage substantial quantities of image data, precisely discern and categorize diverse defect types, and function seamlessly in real-time, aligning with the rapid production pace characteristic of industrial settings.

The objective of this study is to provide valuable insights for decision-making regarding hardware and methods for object detection systems. If this study achieves its goals and validates the effectiveness of a particular solution, it can pave the way for the development of more economical and space-efficient alternatives. Such solutions could potentially replace conventional desktop systems, offering enhanced efficiency and cost-effectiveness. Hence, the main contribution of this paper can be outline as follows:

i. Implementation of object detection algorithm on Raspberry Pi to detect the various kinds of defect in metallic surface.
ii. The proposed work is also been compared using MobileNetv2 and You Look Only Once (YOLOv3) in term of probability and confidence level of detection.

The overview of this paper is as follows: Section 2 explains some related works related to the proposed work. Section 3 describes the proposed methods and algorithms used. Section 4 analyses the performance and analysis of experimental result. Lastly, section 5 concludes the finding of this work.

2. Related Works

In the realm of detecting defects on metallic surfaces, attaining heightened precision and accuracy poses a significant challenge [6]. This study introduces a novel approach to enhance the accuracy of surface defect detection by modifying the architecture of the YOLOv3 algorithm model. This research endeavour to elevate the effectiveness of efficient and accurate surface defect detection systems for metallic surfaces. Defects present on metal materials, encompassing issues like plaques, cracks, pitting, and inclusions, exert adverse effects on product quality and longevity. Some researchers have augmented the Faster Region Convolutional Neural Network (R-CNN) by integrating Region of Interest (RoI) align to enhance the identification of minute defect intricacies, while others have employed the Single Shot Detector (SSD) model coupled with image pre-processing techniques to achieve notable average accuracy in defect detection. Nonetheless, challenges persist in accurately detecting small defects and addressing steel surfaces characterized by lacklustre or uneven colouring. In this study, an enhanced iteration of the YOLOv3 algorithm has been devised, incorporating improved clustering techniques, a dedicated feature layer, and a spatial pyramid pooling module. This advanced iteration attains a superior average accuracy compared to the original YOLOv3 algorithm. Notably, enhancements were particularly evident in detecting scratches and plaques, underscoring the method's efficacy within real-world manufacturing settings.

Another work has also done for an enhanced technology for detecting defects on metallic surfaces by leveraging the YOLO model [7]. The proposed methodology amalgamates the shallow and deep attributes of the Darknet-53 neural network, culminating in the creation of a fresh scale feature layer that amplifies the identification of minor imperfections on metallic surfaces. Furthermore, the selection of anchor boxes is optimized through the utilization of the K-Means clustering algorithm, leading to refined positional precision. Empirical assessments carried out on the Tianchi dataset underscore the superior performance of the modified YOLO model, attaining an average detection accuracy of 75.1% and surpassing the baseline YOLOv3 model. Remarkably, the proposed technique not only showcases improved accuracy but also exhibits considerable advantages in terms of detection speed. To ensure the capture of high-quality images within authentic production settings, this study extensively evaluates and contrasts various options for image acquisition equipment, taking into consideration cost-effectiveness and operational performance. By judiciously selecting the most suitable equipment, the defect detection system is engineered for optimal efficacy and efficiency. When the improved YOLOv3 model is coupled with meticulously selected high-quality images, the resultant system is capable of achieving real-time defect detection while upholding a commendable level of accuracy.

YOLOv3 is also been applied for surface defect detection from different pixel level segmenting methods [8]. Due to low efficiency of conventional detectors to generate region proposals by sliding boxes, pixel level is proposed. YOLOv3 efficiently extracts crucial details about defect locations and class labels with remarkable precision. This information proves adequate for surface defect inspection while concurrently enhancing computational efficiency. To delve deeper into the YOLOv3 structure, optimizations in loss function and a pruning strategy have been implemented in the original YOLOv3. The degree of pruning is determined by a balance between detection accuracy and computational efficiency. In our experimentation, we conducted a comparative analysis of the modified YOLOv3 against various cutting-edge techniques. The results demonstrate that the enhanced YOLOv3 outperforms these alternatives in terms of performance across six surface defect types within the DAGM 2007 datasets.

Surface defects significantly influence the quality of steel products [9]. Swift online detection of steel surface defect images has gained traction among scholars worldwide. Research on steel surface defect recognition technology holds not only theoretical significance but also practical application potential. Addressing the limitations of existing metallic surface defect detection methods notably low detection efficiency, limited applicability, and intricate processing steps. This study presents a real-time defect detection approach grounded in an enhanced YOLOv3 algorithm. In this study, MobileNet is employed as the underlying network to reduce parameter count and elevate network detection efficiency. Recognizing the substantial disparities among distinct defect types on steel surfaces, the paper introduces a hybrid attention mechanism module into the network to acquire a more expansive receptive field. This module incorporates both HaloNet and SENet components. HaloNet captures features with a wide receptive field, while SENet adjusts channel weights to yield superior detection outcomes. Experimental findings substantiate that the proposed method outperforms the current state of the art when evaluated against the NEU-DET datasets.

Deep learning method using YOLOv5 has also been applied in detection of pin solder joints in surface mount technology [10]. High similarity of defect samples and standard samples in solder joint image makes the process of identifying of defect types is challenging. Outstanding performance of detection, YOLOv5 has known as powerful in detection of small objects while highly efficient than YOLOv8. Hence, the authors integrate YOLOv5 with Cascade Shuffle Space to Depth (CSSD) to improve the losses of local feature information during feature extraction while reducing the model parameter size. To enhance the positioning ability of backbone network and misses of detection rate, mechanism module namely Coordinate Attention (CA) is proposed. In addition, well known clustering method, k-means has also been applied to optimize anchor box sizes. As a result, the improvement has obtained with precision of 12.2% and mAP of 9% compared to original YOLOv5 model. Another reported work has also been utilized deep learning YOLO model in manufacturing technology [11]. Traditional machine learning models found to be difficult to detect complex defect types with noisy environments. Hence, deep learning YOLOv4 models has introduced to recognize various kind of defects using wire and arc additive manufacturing (WAAM) defect datasets. The proposed models have improved about 29% of mAP with least 42 frames per second for classifying four types of defects including; weld, surface pore, groove and slag inclusion.

3. Methodology

The experiment is divided into 2 sections; MobileNetv2 and YOLOv3. The hardware development phase encompasses the establishment of the Raspberry Pi system and its integration with the webcam for real-time implementation. The model training phase will commence with data pre-processing, followed by the training and testing of the model. If the model demonstrates satisfactory accuracy, it will subsequently be deployed onto the Raspberry Pi.

3.1 Dataset

This project employs the GC10-DET metallic surface defect datasets for model training (Website: https://github.com/lvxiaoming2019/GC10-DET-Metallic-Surface-Defect-Datasets). GC10-DET is an industrial metallic surface-defect dataset collected from a real industry that consists of 3570 grey images with size of 2048 x 1000. Acquired from an actual industrial context, this dataset incorporates ten distinct surface defect categories such as punching, weld line, crescent gap, water spot, oil spot, silk spot, inclusion, rolled pit, crease, and waist folding as shown on Figure 1. These anomalies manifest on steel sheet surfaces. The datasets comprise 3570 grayscale images accompanied by 2280 XML label files. 10 types of metallic surface defections are shown with the bounding box and respective labels. For instance, in the example shown at the bottom left corner, three waist folding defects are accurately labeled. In this project, three types of defects are chosen, including crescent gap, inclusion and oil spot. Therefore, 639 images with the corresponding label files are used to train the model.

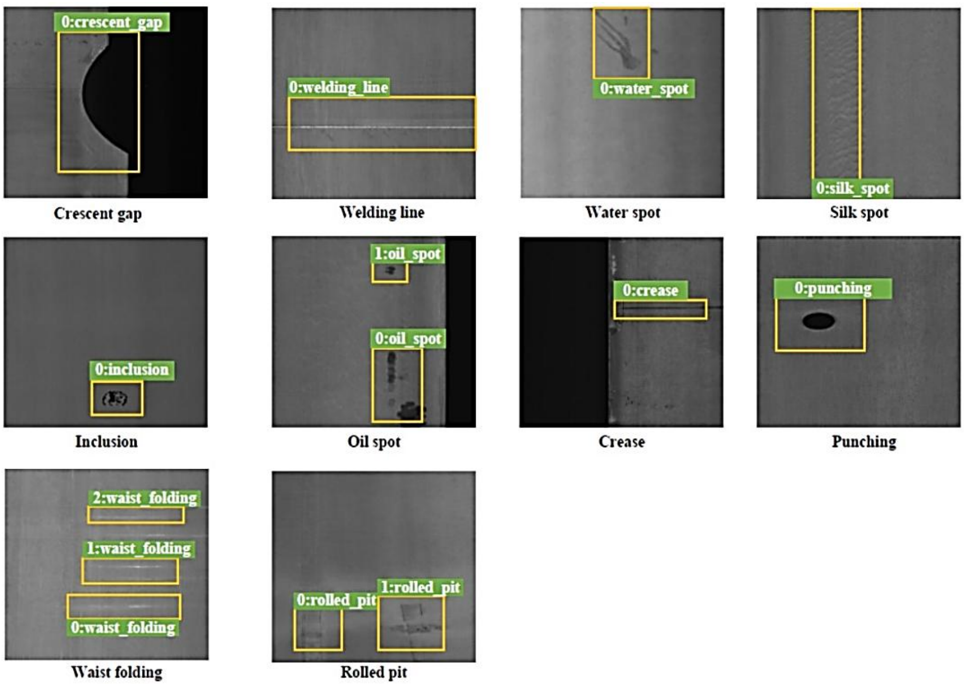


Fig. 1. Example of defect surface

3.2 Data Pre-Processing

Given the utilization of the YOLOv3 tiny algorithm for model training, it becomes imperative to transform the original label XML files within the datasets into the TXT file format. This TXT file must encompass pertinent metadata: object ID, centre coordinates (x and y), width, and height. Additionally, an auxiliary TXT file labeled "classes.txt" is generated, incorporating a comprehensive list of all annotated classes from the datasets. In this specific instance, the "classes.txt" file will feature the class oil spot, crescent gap, and inclusion. The object ID corresponds to numerical identifiers associated with the defect classes enumerated in the "classes.txt" file, as outlined in Table 1. Both centre x and centre y are normalized within the range of 0 to 1, representing the centre point of the bounding box. Then, the width and height denote the size of the bounding box as depicted in Figure 2.

Table 1
Object class ID

Object ID	Classes
0	Oil spot
1	Crescent gap
2	Inclusion

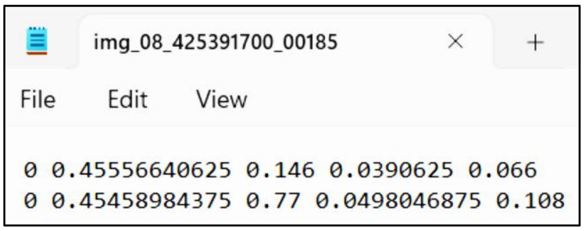


Fig. 2. Example of defect surface

3.3 YOLO

YOLO re-imagines object detection as a singular regression task. YOLO processes entire images in one go, directly predicting bounding box coordinates and class probabilities. This results in swifter and more efficient detection capabilities, as it eliminates the necessity for region proposals and facilitates training on complete images. Compared to traditional object detection methods, YOLO models offer a host of advantages. A prominent strength of YOLO is its speed and efficiency. It streamlines the detection process by treating it as a prediction task. During testing, YOLO simply runs the neural network on a new image to generate predictions for detection. Furthermore, YOLO achieves a mean average accuracy that is more than twice that of comparable real-time object detection systems. Another notable benefit of YOLO is its comprehensive methodology for image analysis. It employs a solitary neural network architecture that concurrently forecasts object bounding boxes and associated class probabilities.

This approach facilitates real-time object detection by handling the entire image in one pass, obviating the requirement for region proposal techniques employed in alternate object detection methods [12]. In contrast to sliding window and region proposal-centred approaches, YOLO takes into account the complete image during both training and testing phases. Moreover, YOLO demonstrates remarkable proficiency in acquiring generalized object representations. When tested on artistic images and trained using real-world visuals, YOLO surpasses leading detection methods like deformable parts model (DPM) and R-CNN. Its aptitude to exhibit strong generalization across diverse scenarios and manage unforeseen inputs minimizes the risk of faltering when confronted with unfamiliar circumstances. This adaptability and resilience render YOLO an exceedingly potent and versatile model for object detection [13] in various industries including agricultural [14], medical [15] and human activity using vision sensors [16].

3.4 MobileNet

The proposed approach leverages MobileNetV2 [17] as its foundational model. MobileNetV2 is an evolution of MobileNetV1 [18], featuring the inclusion of inverted residual with linear bottleneck modules. The MobileNet architecture is based on the concept of depthwise separable convolution. Unlike the standard 2D convolution that directly processes all input channels to produce a solitary output channel by convolving across the depth dimension (channel), depthwise convolution segregates input channels and filter channels, convolving them individually. The output channels thus generated are reassembled after convolution. In the case of separable depthwise convolution, these stacked output channels are further filtered using a 1×1 convolution, referred to as pointwise convolution. This amalgamates the stacked output channels into a singular channel. Depthwise separable convolution yields equivalent outputs to standard convolution but boasts increased efficiency by reducing the number of parameters involved in the process [17].

3.5 Raspberry Pi

The Raspberry Pi is a compact computer, roughly the dimensions of a standard deck of cards. Currently, the market offers five distinct Raspberry Pi models: Model B+, Model A+, Model B, Model A, and the Compute Module (which is presently exclusively accessible as a component of the Compute Module development kit). It's important to note that all these models share the same System on Chip (SoC) called BCM2835, but they may differ in other hardware specifications. The Raspberry Pi Camera Module v2 is equipped with a Sony IMX219 8-megapixel sensor, a notable improvement compared to the original camera, which featured a 5-megapixel OmniVision OV5647 sensor. This upgraded camera module is utilized for capturing high-definition videos and images. In terms of programming languages, Python is the predominant and well-supported programming language on the Raspberry Pi. However, the Raspberry Pi ecosystem also offers compatibility with several other programming languages, including but not limited to BASIC, C, C++, Java, Perl, and Ruby. This flexibility empowers users to develop software and applications using their preferred programming language. The Raspberry Pi's adaptability extends beyond programming languages; its hardware specifications play a pivotal role in its versatility. Models like the Raspberry Pi Model B+ feature components such as the Broadcom BCM2835 SoC processor, 512MB of RAM, Videocore 4 GPU, MicroSD card slot, Ethernet port, USB ports, HDMI and audio/video jacks, GPIO header, and more. These diverse features make the Raspberry Pi suitable for a wide spectrum of projects, encompassing domains like robotics, home automation, media players, data logging, and applications in the Internet of Things (IoT).

4. Results

In this section, we divided the experimental results into 2 parts; MobileNetv2 and YOLOv3. The class silk spot boasts the highest frequency within the datasets, with a total count of 650 occurrences. Following closely, the second highest count corresponds to the water spot class, total of 289 instances. Welding line appears 273 times, crescent gap has 226 instances, punching hole tallies 219 occurrences, inclusion appears 216 times, oil spot is noted 204 times, waist folding has 146 occurrences, crease registers 52 instances, and rolled pit records the least count of 31 instances within the datasets. There are some samples of plots for different classes of defects from the original datasets. A sample of plot for defect class silk spot, punching holes and crescent gap as shown in Figure 3. In this work, we only detect and identify the defect types without including any other criteria such as the number of defects found, the size of defects and locations on the metallic surface.

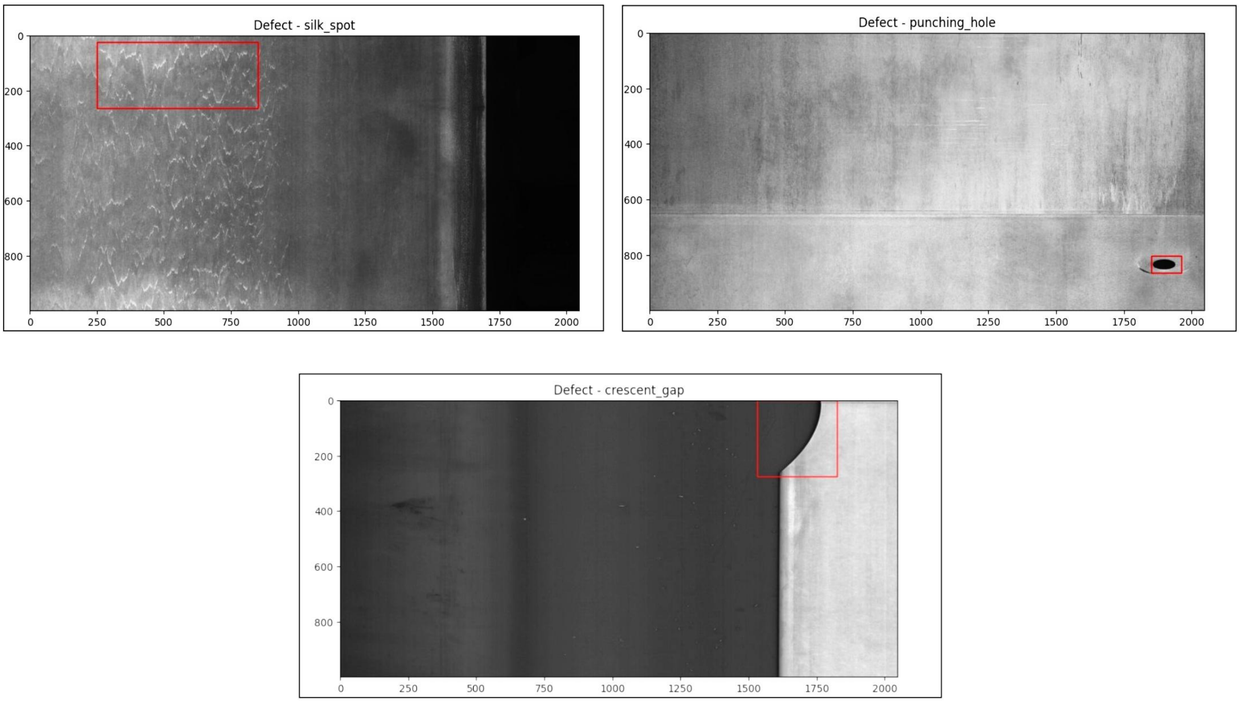


Fig. 3. Area of silk spot, punching holes and crescent gap defects

There are few parameters need to be defined accordingly before the training process begin. Due to the limitation of the Raspberry Pi and to ensure all the samples is fairly evaluated, the original images is resized 256 x 512. After the images is resized, the entire datasets are split into training and validation. 80% of sample is used for training while 20% is reserved for validation with batch size of 32. The convolutional layer used as an input layers with size of 3, strides of 2 and rectified linear function (ReLU) is used as activation functions. The new unseen image is used for testing the model in Raspberry Pi for real-time implementation. Figure 4 shows the number of distribution count for training subsets.

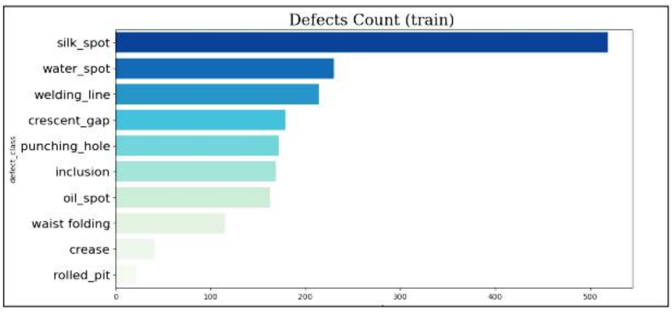


Fig. 4. Number of defect count based on 10 categories

Figure 4 illustrates the loss values and accuracy for both training and validation using MobileNetv2. Loss serves as a metric for evaluating the model's performance, assessing the disparity between predicted outputs and actual ground truth values. The chart's line illustrates the trend of decreasing loss across epochs. A consistent decline in loss indicates the model's capacity to learn and enhance its predictive capabilities. Meanwhile for the accuracy, the chart's line reflects the trend of accuracy increasing throughout epochs. A significant rise in accuracy, leveling off at a high plateau, indicates the model's learning process and its improved accuracy in predictions. As we can see from Figure 5, the correct prediction rates for both silk spot and crescent gap stand above

98%, representing the highest values. Welding line exhibits a prediction proportion of 63%, while inclusion archives 88%. Waist folding attains a prediction rate of 93%, whereas crease displays the lowest proportion below 60%. Water spot demonstrates a rate of 79%, punching holes records 70%, oil spot and rolled pit each exhibit a proportion of 88% and 67%, respectively.

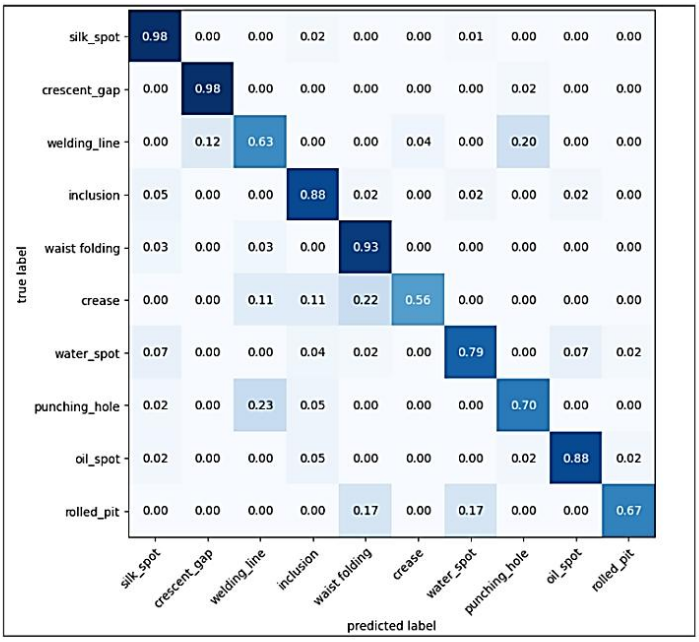


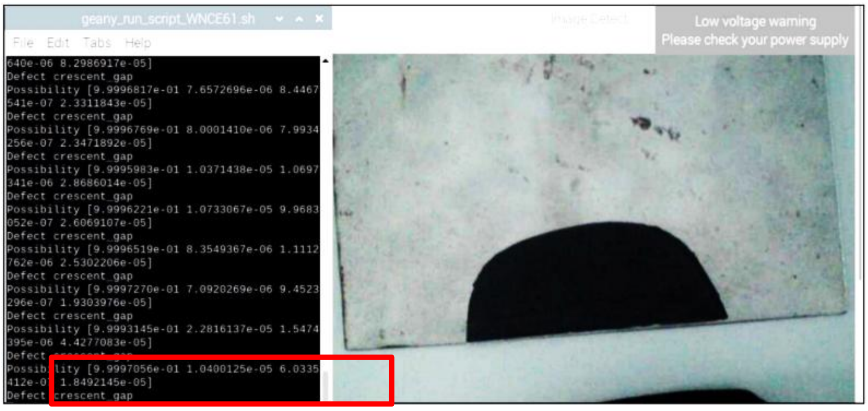
Fig. 5. Confusion matrix of training MobileNetv2 model for 10 classes

Due to the resource constrain of Raspberry Pi, we reduce the number of classes to 4 classes of defect; crescent gap, punching hole, oil spot and rolled pit. As a result, we manage to achieve 100% of accuracy for oil spot and rolled pit followed by crescent gap with prediction rate of 95%. Yet, punching hole recorded slightly lower with 79% of accuracy in average. Table 2 depicts the training accuracy using MobileNetv2 for 4 classes.

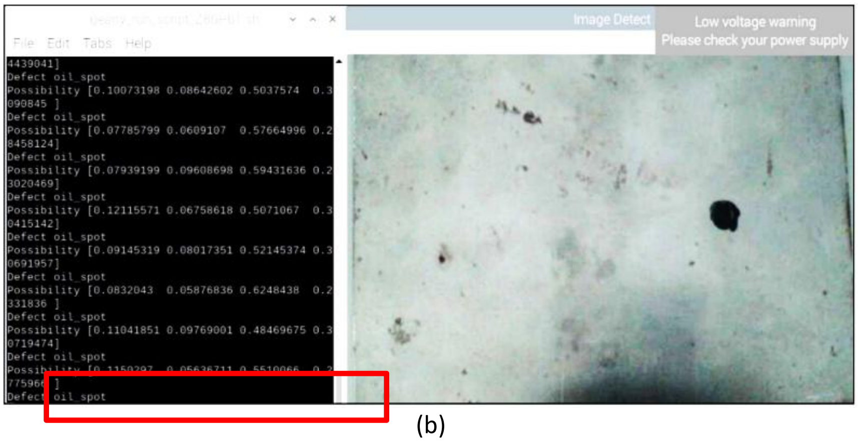
Table 2
Training accuracy using MobileNetv2

Classes	Accuracy (%)
Crescent gap	95
Punching hole	79
Oil spot	100
Rolled pit	100

The trained model later is deployed into Raspberry Pi for evaluating our testing subset. In this part, we integrate the Raspberry Pi with web camera for evaluating real-time defect detection. Figure 6(a) illustrates a metal object bearing the defect class crescent gap. The model's prediction output highlights its ability to process one frame per second. It systematically evaluates potential classes, ultimately providing an accurate prediction of the defect class with highest probability or class 1 (crescent gap) above 9.9. We also tested real-time prediction of oil spot sample. The model recorded highest probability with above 0.55 in predicting oil spot as shown in Figure 6(b).



(a)



(b)

Fig. 6. (a) Prediction class of crescent gap (b) Prediction class of oil spot

The second evaluation is to perform experiment using YOLOv3 model. We trained the model to learn the defect on the metallic surface similar to the MobileNetv2. In YOLOv3 model, Darknet53 is used as a backbone with the batch size of 64. Figure 7 shows the accuracy of detection for various kinds of defects accordingly. In this part, we used confidence level as performance indicators by measuring on the intersection of bounding box from actual object with the predicted bounding box. The confidence level above 0.50 is acceptable to notify the model is able to detect of the object presence, otherwise it is considered unsuccessful.

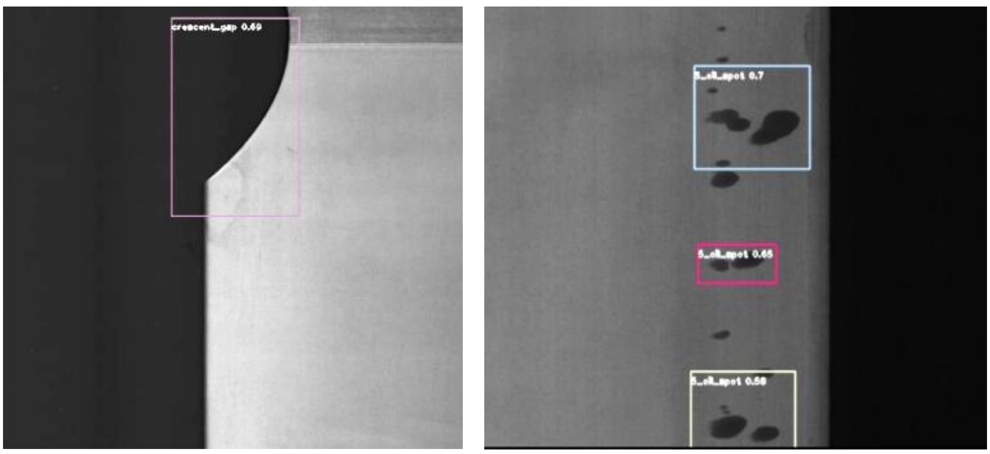
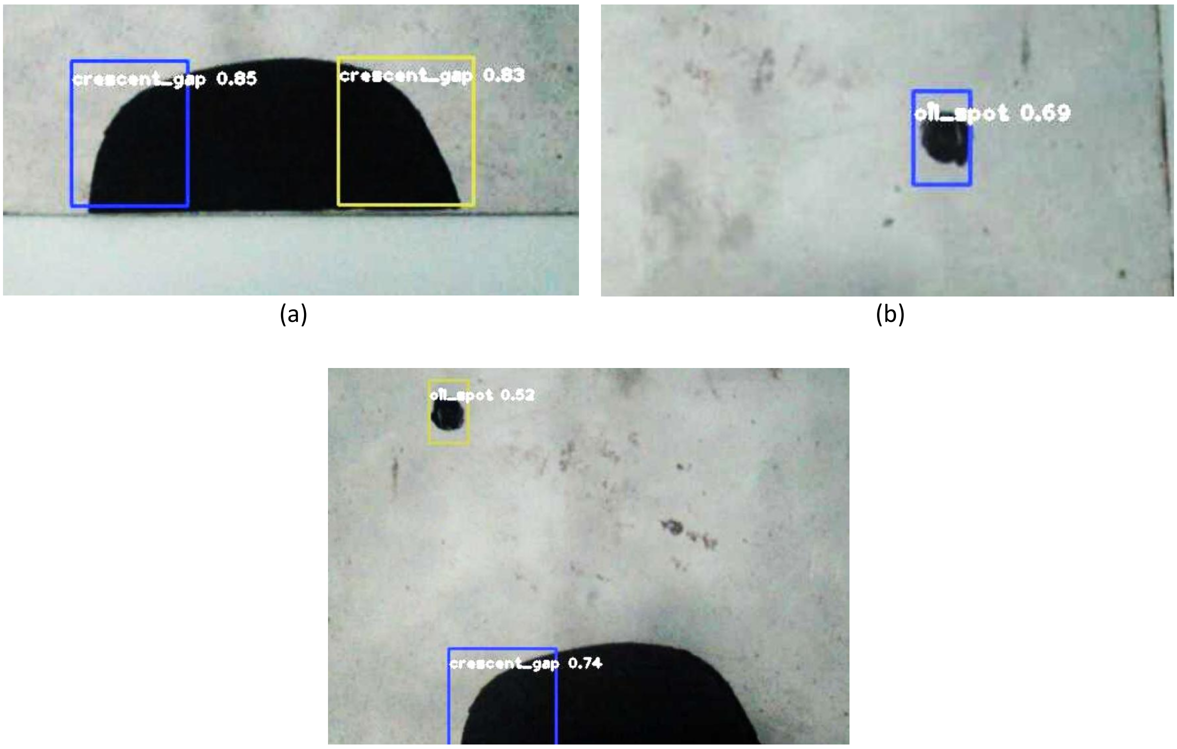


Fig. 7. Detection of crescent gap (top) and oil spot (down)

It can be observed the model's adeptness in correctly categorizing the defect the crescent gap class, precisely delineating the bounding box around the pertinent defect area with a confidence level of 0.69. The sample of oil spot is also been evaluated. As a result, the model is detected 3 bounding boxes to indicate the defect. All three bounding boxes recorded above 0.58 to define defective parts from the surface. However, the small spot is unable to be detected since the limitations of YOLO model is incapable to detect the small object even though the YOLO versions model able to record high efficiency of detection.

The trained model later is embedded into Raspberry Pi to test its capabilities of detection in real-time performance. As we can observed in Figure 8(a), the crescent gap is been detected for both left and right positions of the defect with confidence level above 0.83. The oil spot is also could be detected with 0.69 confidence as stated in Figure 8(b). For the third samples, we tested for more one type of defect on the surface. The output of the detection shows that the model can accurately classify more than one defect classes at one time and detect the parts of defect with 0.74 (crescent gap) and 0.52 (oil spot) confidence respectively as shown in Figure 8(c).



(c)

Fig. 8. Detection in real-time performance (a) Crescent gap (b) Oil spot (c) Crescent gap and oil spot

In summary, we are able to conclude the objectives of the proposed work has achieved. We conducted a comparative analysis with prior research that employed the identical dataset. The previous researchers incorporated diverse defect types in their experiments, such as punching, weld line, crescent gap, water spot, oil spot, silk spot, inclusion, rolled pit, crease, and waist folding [19]. It is noteworthy that their experiment was executed using the YOLO algorithms on a workstation. Table 3 illustrates the juxtaposition between our proposed approach and the preceding research.

The results clearly indicate that the majority of defect classes achieved high detection accuracy without relying on a micro-controller. However, challenges were observed in effectively detecting oil spot and rolled pit, primarily attributed to the high similarity in defect characteristics. Despite the proposed method not delivering exceptional overall performance, we conducted an experiment to assess its real-time recognition capability for various metallic surface defects on a Raspberry Pi. Leveraging the Raspberry Pi platform provides an economical solution tailored for industrial environments, characterized by its compact size, low energy consumption, and computational capabilities.

Table 3
Comparison with previous work using GC10-DET metallic surface defect datasets

Classes	SSD	Faster-RCNN	YOLOv2	YOLOv3	Proposed [19]	Our method
Crescent gap	0.861	0.875	0.819	0.752	0.848	0.840
Punching hole	0.860	0.899	0.725	0.836	0.900	0.790
Oil spot	0.612	0.653	0.403	0.329	0.622	0.690
Rolled pit	0.105	0.364	0.018	0.036	0.364	0.450

5. Conclusions

In summary, this project has successfully fulfilled its objectives by developing an effective metallic surface defect detection using MobileNetv2 and YOLOv3 on the Raspberry Pi platform. The endeavour encompassed system design, development, and programming, with a focal point on real-time defect detection to bolster manufacturing procedures. A significant achievement of this undertaking is the system's capacity to accurately and efficiently identify defects on metallic surfaces, thereby contributing to enhanced product quality and decreased labour expenses. The devised system showcases numerous benefits. Firstly, it facilitates real-time defect detection, enabling swift identification and immediate corrective measures to mitigate potential production setbacks. Secondly, by automating the defect detection process, the system reduces reliance on time-consuming and error-prone manual inspections. This streamlines quality control processes, ensuring consistent and dependable defect identification. Lastly, the utilization of the Raspberry Pi platform offers an economical solution suited for industrial settings, characterized by its compact size, low energy consumption, and computational capabilities.

For future enhancements, it is advisable to explore advancements in materials and processors to bolster the system's performance and accuracy. Additionally, further research could focus on refining the system's design and algorithms, yielding more efficient defect detection and classification. However, it is crucial to consider constraints such as time, cost, and availability of equipment when planning for future improvements. Overall, this metallic surface defect detection system holds immense potential in enhancing manufacturing processes and quality control across diverse industries, ultimately leading to augmented productivity and heightened product dependability.

Acknowledgement
The authors would like to thank Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM) under matching grant number INDUSTRI(MTUN)/ENDSTRUCT/ 2021/FKEKK/I00057.

References
[1] Zhao, Haili, Zefeng Yang, and Jia Li. "Detection of metal surface defects based on YOLOv4 algorithm." In *Journal of Physics: Conference Series*, vol. 1907, no. 1, p. 012043. IOP Publishing, 2021. https://doi/org/10.1088/1742-6596/1907/1/012043
[2] Chigateri, Keerthana B., and Ajit M. Hebbale. "A steel surface defect detection model using machine learning." *Materials Today: Proceedings* (2023). https://doi.org/10.1016/j.matpr.2023.04.646

[3] He, Hongyang, Mingang Yuan, and Xiushan Liu. "Research on surface defect detection method of metal workpiece based on machine learning." In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, p. 881-884. IEEE, 2021. https://doi.org/10.1109/ICSP51882.2021.9408778

[4] Wang, Shuai, Xiaojun Xia, Lanqing Ye, and Binbin Yang. "Automatic detection and classification of steel surface defect using deep convolutional neural networks." *Metals* 11, no. 3 (2021): 388. https://doi.org/10.3390/met11030388

[5] Tan, Chi Wee, Joren Mundane Antequisa Pacaldo, Wah Pheng Lee, Gloria Jennis Tan, Siaw Lang Wong, and Jun Kit Chaw. "Automated metal surface defect detection." In *International Conference on Machine Learning and Intelligent Communications*, p. 39-49. Cham: Springer Nature Switzerland, 2022. https://doi.org/10.1007/978-3-031-30237-4_4

[6] Shao, Xianxin, XiaoJun Xia, and Jia Song. "Design and research of metal surface defect detection based on machine vision." In *2022 11th International Conference of Information and Communication Technology (ICTech))*, pp. 406-410. IEEE, 2022. https://doi.org/10.1109/ICTech55460.2022.00087

[7] Xu, Yiming, Kai Zhang, and Li Wang. "Metal surface defect detection using modified YOLO." *Algorithms* 14, no. 9 (2021): 257. https://doi.org/10.3390/a14090257

[8] Wang, Zhihui, Houying Zhu, Xianqing Jia, Yongtang Bao, and Changmiao Wang. "Surface defect detection with modified real-time detector YOLOv3." *Journal of Sensors* 2022, no. 1 (2022): 8668149. https://doi.org/10.1155/2022/8668149

[9] Zheng, Jiexin, Zeyang Zhuang, Tao Liao, and Lihong Chen. "Improved Yolo V3 for steel surface defect detection." In *International Conference on Computer Engineering and Networks*, pp. 729-739. Singapore: Springer Nature Singapore, 2022. https://doi.org/10.1007/978-981-19-6901-0_75

[10] Ang, Li, Raseeda Hamzah, Siti Khatijah Nor Abdul Rahim, Gao Yousheng, and Khyrina Airin Fariza Abu Samah. "CSSD-YOLO: A modified YOLOv5 method for solder joint defect detection." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 31, no. 3 (2023): 249-264. https://doi.org/10.37934/araset.31.3.249264

[11] Li, Wenhao, Haiou Zhang, Guilan Wang, Gang Xiong, Meihua Zhao, Guokuan Li, and Runsheng Li. "Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing." *Robotics and Computer-Integrated Manufacturing* 80 (2023): 102470. https://doi.org/10.1016/j.rcim.2022.102470

[12] Sozzi, Marco, Silvia Cantalamessa, Alessia Cogato, Ahmed Kayad, and Francesco Marinello. "Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms." *Agronomy* 12, no. 2 (2022): 319. https://doi.org/10.3390/agronomy12020319

[13] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. 2016. https://doi.org/10.48550/arXiv.1506.02640

[14] Zainudin, MN Shah, MSS Shahrul Azlan, L. L. Yin, WH Mohd Saad, M. I. Idris, Sufri Muhammad, and M. S. J. A. Razak. "Analysis on localization and prediction of depth chili fruits images using YOLOv5." *International Journal of Advanced Technology and Engineering Exploration* 9, no. 97 (2022): 1786. https://doi.org/10.19101/IJATEE.2021.876501

[15] Ismail, Mohd Muzafar, MN Shah Zainudin, Rosman A. Rahim, Maizatul Alice, and N. Mohd Kassim. "An investigation of electromagnetic field effect on a human skin cell using numerical method approaches." In *2011 IEEE International RF & Microwave Conference*, pp. 178-182. IEEE, 2011. https://doi.org/10.1109/RFM.2011.6168724

[16] Zainudin, MN Shah, Md Nasir Sulaiman, Norwati Mustapha, and Thinagaran Perumal. "Monitoring daily fitness activity using accelerometer sensor fusion." In *2017 IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 35-36. IEEE, 2017. https://doi.org/10.1109/ISCE.2017.8355540

[17] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 4510-4520. 2018. https://doi.org/10.48550/arXiv.1801.04381

[18] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "MobileNets: Efficient convolutional neural networks for mobile vision applications (2017)." *arXiv preprint arXiv:1704.04861* 126 (2017). https://doi.org/10.48550/arXiv.1704.04861

[19] Lv, Xiaoming, Fajie Duan, Jia-jia Jiang, Xiao Fu, and Lin Gan. "Deep metallic surface defect detection: The new benchmark and detection network." *Sensors* 20, no. 6 (2020): 1562. https://doi.org/10.3390/s20061562