IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 December 2023, accepted 17 January 2024, date of publication 30 January 2024, date of current version 21 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359639

RESEARCH ARTICLE

Enhancing EfficientNet-YOLOv4 for Integrated Circuit Detection on Printed Circuit Board (PCB)

TAY SHIEK CHI[1], MOHD NADHIR AB WAHAB[1], (Member, IEEE), AHMAD SUFRIL AZLAN MOHAMED[1], MOHD HALIM MOHD NOOR[1], KHAW BENG KANG[2], LIM LAY CHUAN[2], AND LIAU WEI JIE BRIGITTE[1]

[1]School of Computer Sciences, Universiti Sains Malaysia, Minden, Penang 11800, Malaysia
[2]SanDisk Storage Malaysia Sdn. Bhd., Seberang Perai Selatan, Penang 14100, Malaysia

Corresponding author: Mohd Nadhir Ab Wahab (mohdnadhir@usm.my)

This work was supported in part by the Collaborative Research in Engineering, Science and Technology (CREST) Industry Graduate Research Assistant Scholarship Program (CREST-i-GRASP), Universiti Sains Malaysia (USM), and in part by SanDisk Storage Malaysia Sdn. Bhd under CREST Research and Development Project P08C1-20 "Image Data Analytics for Industry 4.0" Grant.

ABSTRACT Ensuring the quality and functionality of printed circuit boards (PCBs) during manufacturing requires precise, automated visual inspection. Detecting integrated circuits (ICs) on PCBs poses a significant challenge due to diverse component sizes, types, and intricate board markings that complicate accurate object detection. This study addresses this challenge by proposing an enhanced EfficientNet-YOLOv4 algorithm tailored explicitly for the IC detection of PCBs. Numerous modifications are integrated into YOLOv4, with the replacement of its original backbone by a robust feature extraction network, EfficientNetv2-L, and meticulous hyperparameter tuning, including variations in loss functions, anchor size configurations, and other training techniques. The methodology further incorporates diverse data augmentation techniques to enrich the training dataset and enhance the model's generalization ability. Extensive experiments conducted in this study showed the efficacy and robustness of the algorithm in handling complex PCB layouts and varying lighting conditions, outperforming existing PCB inspection models. The proposed method, EfficientNetv2-L-YOLOv4, achieved an impressive F1-score of 99.22 with an inference speed of 0.14 s per image. The proposed method also performed well compared to EfficientNet-B7-FasterRCNN and the original YOLOv4; it attains an F1-score of 98.96 and an inference speed of 0.10 s per image (with a batch size of 4). These results highlight the significance of effective feature extraction networks for object detection. Beyond addressing IC detection challenges, this algorithm advances the fields of computer vision and object detection. The implementation of EfficientNetv2-L-YOLOv4 in real manufacturing scenarios holds promise for automating component inspections and potentially eliminating the need for human intervention.

INDEX TERMS Automated visual inspection, feature extraction network, object detection, printed circuit board (PCB).

I. INTRODUCTION

Industry 4.0 marks a transformative era in which the Internet of Things (IoT) and artificial intelligence (AI) redefine supply chain automation, especially within manufacturing, leveraging artificial intelligence and machine learning to elevate global supply and value chains [1]. Technological advancements enable the digital transformation of factories by automating industrial processes, aiming for autonomous operations and attaining high-quality electronic production equipment. Machine vision is crucial in modern electronics manufacturing, primarily manifesting in four key aspects:

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

measurement, inspection, identification, and positioning [2]. One notable application is printed circuit board (PCB) inspection, where intelligent vision technology is employed to ensure precision and product quality.

Printed circuit boards serve as cores for electronic devices, interconnected housing circuits, and electronic components. Traditional manual visual inspection, prone to inefficiencies and errors owing to human limitations, requires non-contact automation methods, particularly in PCB assembly (PCBA) [3]. Automatic Optical Inspection systems (AOIs) have emerged as pivotal machine vision applications to ensure PCB quality and streamline inspection processes, thereby alleviating manual inspection challenges [4]. AOI systems advance data acquisition by capturing high-resolution images of PCBs using cameras. These images are meticulously analyzed against design specifications to detect defects, providing a robust dataset for quality assessment. Alternative inspection methods for PCBs, including Automated X-ray Inspection (AXI), Infrared Thermography (IRT), and Acoustic Micro Imaging (AMI), are employed for quality assurance.

Accurate component detection is pivotal in automating PCB production monitoring, specifically in addressing critical manufacturing defects such as component shifts or missing parts within Surface Mount Technology (SMT) pick-and-place processes [2], [5]. The enhancement of automated PCB inspection tools is imperative for effectively tackling these issues, optimizing efficiency, and enabling swift, precise, and early fault detection across all stages of production. Additionally, pinpointing the exact component location not only aids in defect inspection but also facilitates character identification of PCB components and supports the recycling process of the PCB.

The detection and localization of ICs on PCBs remains a formidable challenge for automated inspection systems. This difficulty arises from the complex variability in component sizes, orientations, and layouts encountered during inspections. Object detection, a fundamental facet of computer vision, relies on machine learning or deep learning methodologies to extract meaningful insights from images. It encompasses two integral components: image classification and localization, both of which are vital for the identification and precise positioning of PCB components.

Object detection in PCB inspection presents substantial opportunities for enhancement, particularly in critical aspects such as feature learning, backbone architecture, and proposal generation [6]. Challenges persist in effectively handling feature-scale issues and mastering multiscale feature learning, both essential for accurately identifying diverse ICs. The pursuit of a detection-aware backbone architecture tailored explicitly for object detection has emerged as a pivotal research focus [6]. However, identifying the most suitable backbone architecture within PCB datasets remains a significant challenge, impacting the precision and complexity of object detection tasks. Achieving a balance between speed and accuracy necessitates adaptive multilevel features and a well-designed backbone architecture [7]. As current feature extraction networks face challenges in capturing intricate details across diverse PCBs, there is a growing interest in exploring the direct learning of backbone architectures from datasets [6]. Significant strategies involve optimizing backbone architectures such as Neural Architecture Search (NAS) within Auto Machine Learning (AutoML) or adapting existing architectures such as EfficientNet to tailor them to specific object-detection tasks.

Moreover, hyperparameter settings in machine learning represent predetermined choices that significantly influence the behavior, complexity, and speed of the learning process, and these values must be carefully chosen to achieve optimal performance [8], [9]. The underexplored realm of hyperparameter tuning in machine learning remains mainly uncharted, resulting in a conspicuous lack of systematic analyses of parameter tuning practices in academic research. Consequently, there is a need for the systematic exploration and refinement of these configurations to enhance the performance of object detection models.

This study explicitly targets the detection of integrated circuits on a PCB, excluding their pins or soldering parts. The focus of this research is on implementing the feature extraction network and fine-tuning the training settings, aiming to further improve the accuracy of object detectors. The contributions of this study are as follows:

- Enhance object detection performance, particularly for ICs on PCBs, by gaining valuable insights into backbone architecture development and selection.
- Optimize model configurations, including variations in loss functions, anchor sizes, and training techniques.
- Explore image-augmentation techniques to improve the generalization ability of the model.

The remainder of this paper is organized as follows. Section II provides a review of the relevant literature on object detection algorithms, with a particular focus on those applied to PCB inspections. Section III outlines the methodology, including details on the data collection process, and discusses the selection and modification of the proposed algorithm. Comprehensive experiments to evaluate the performance of the proposed method are presented in Section IV. Finally, Section V summarizes the key findings of this study and offers recommendations for future research in this field.

II. RELATED WORKS

In PCB assembly, object detection techniques prove invaluable for identifying and classifying various types of electrical components, including resistors, capacitors, and integrated circuits. These techniques are equally valuable for detecting common defects like soldering issues (open circuits, excess solder) and component irregularities (missing or misaligned elements) on the PCB. This exploration delves into different categories of deep learning-powered object detectors and neural network-based methods, underscoring their significance in PCB inspection. The focus of the reviewed techniques lies predominantly on PCB component detection, extending to the examination of common electrical components, and incorporates research on PCB defect detection, offering a comprehensive understanding of PCB inspection methodologies. Figure 1 shows the scope of the literature review.

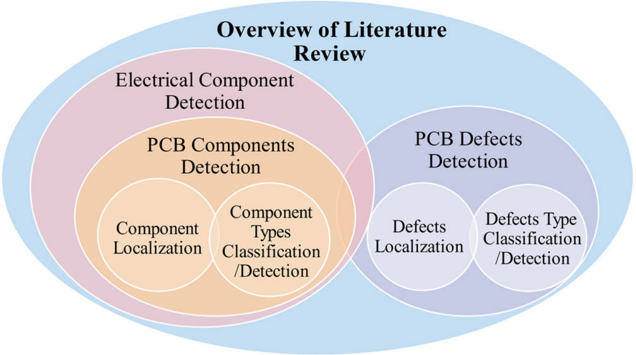

Overview of Literature Review
Electrical Component Detection
PCB Components Detection
Component Localization
Component Types Classification /Detection
PCB Defects Detection
Defects Localization
Defects Type Classification /Detection

FIGURE 1. Overview of the literature review.

A. DEEP LEARNING-BASED OBJECT DETECTORS

Object detection in deep learning is commonly categorized into two main paradigms: two-stage and one-stage methods. Two-stage methods are renowned for their precision in predictions but are often associated with increased computational overhead due to an additional step involving the identification of regions of interest before classification [10]. In contrast, one-stage methods streamline the process by performing object detection in a single step [11]. These methods directly predict class labels and bounding box coordinates for all potential objects within an image, eliminating the need for explicit region proposal generation and offering notable advantages in computational efficiency [2].

Exemplifying two-stage methods are established architectures such as R-CNN (Region-based Convolutional Neural Network), Fast R-CNN, Faster R-CNN, Mask R-CNN (Mask Region-based Convolutional Neural Network), and R-FCN (Region-based Full Convolutional Network) [2]. On the other hand, YOLO (You Only Look Once) and RetinaNet emerge as regression-based deep learning algorithms, classified under the one-stage method [2].

1) ONE-STAGE DETECTOR: YOLO

The YOLO method, based on Convolutional Neural Networks (CNNs), was widely utilized for real-time prediction in PCB assemblies. YOLOv3, distinguished from YOLOv2 primarily by its Feature Pyramid Network (FPN) architecture, excelled in multi-scale prediction and effective small-object detection [12]. In the detection of small surface-mounted devices (SMD) on PCBs, Li et al. [2] proposed enhancements to the YOLOv3 model by introducing a target-sensitive YOLO output layer to prevent the loss of feature information for small components. Addressing the concern of absent components in PCBs, Khare et al. [13] introduced PCB-Fire, a solution involving object detection (using YOLOv3), image subtraction, and pixel manipulation. Silva et al. [14] applied a pre-trained YOLOv3 model, fine-tuned with the publicly available PCB DSLR dataset [15], to detect ICs in waste PCBs, thereby facilitating the recycling process.

In YOLOv3, bounding box predictions depended on the anchor-box concept [2]. However, the performance could be affected by discrepancies between the anchor and target sizes. This issue was addressed by employing K-means clustering, which generated the anchors suitable for the distribution of PCB electronic components based on the size ratio of the target in the training dataset [2], [16]. The YOLOv3 was successfully enhanced by applying K-means clustering to generate 12 anchor boxes in PCB electronic component detection [2].

The backbone networks responsible for feature extraction played a pivotal role in identifying objects within images. In their study, Chen and Tsai [16] replaced the Darknet-53 backbone of YOLOv3 with DenseNet-121 for defect inspection in SMD LED chips, aiming to enhance the efficiency of defect identification. The evolution of object detection algorithms notably elevated YOLOv4 as the preferred choice for PCB detection, surpassing its predecessor YOLOv3. Caliskan and Gurkan [17] successfully employed the YOLOv4 algorithm to detect solder joint defects in assembled PCB production lines. Subsequently, YOLOv4 underwent further improvements and found applications in defect-detection methods for PCB electronic components [18]. Furthermore, the integration of the YOLOv4-tiny algorithm with a Multiscale Attention Module (MAM) proved effective in enhancing the accuracy of electronic component detection [19].

In object detection, the loss function typically encompassed classification loss, confidence loss, and bounding box regression loss—each evaluating distinct aspects of performance. The loss function quantified the disparity between the predicted and actual (ground truth) values and assessed the proximity or dissimilarity between these values. Its primary purpose was to guide the learning process of the model and facilitate parameter updates during optimization. Various loss functions were introduced to object detectors for component detection, including Generalized Intersection over Union (GIoU) [3], Gaussian Intersection of Union (GsIoU) [20], Loss Boosting (LB) [10], and modified binary cross-entropy (BCE) [19].

2) ONE-STAGE DETECTOR: RetinaNet

In an evaluation of PCB analysis methods, Mahalingam et al. [21] explored diverse approaches, such as YOLOv3, RetinaNet-50, and Faster R-CNN. RetinaNet, designed as a one-stage object detector, utilized focal loss for classification and featured a unified network with a backbone and two subnets for classification and box regression tasks. Despite RetinaNet exhibiting the best overall performance among the evaluated models, it faced challenges in distinguishing components resembling ICs. Furthermore, they also introduced a publicly available PCB image dataset, PCB-METAL, encompassing various PCB components [21].

3) TWO-STAGE DETECTOR

Mallaiyan Sathiaseelan et al. [22] introduced ECLAD-Net, an Electronic Component Localization and Detection Network designed for detecting counterfeits and defects in PCB assembly. The ECLAD-Net comprised two stages: the Region Proposal Network (RPN), suggesting regions, and the Similarity Prediction Network (SPN), functioning as a classifier to distinguish between resistors and capacitors. In another approach, Kuo et al. [23] proposed a three-stage object detection pipeline. The RPN identified potential components using bounding boxes, while the SPN addressed imbalanced distributions among various types of PCB components.

Various methodologies rooted in Faster R-CNN emerged for PCB inspection. A specific variant of Faster R-CNN, Inception-v2, exhibited promising performance in localizing PCB components, particularly in identifying absent resistors [24]. EfficientNet found applications in various PCB-related domains. Fan et al. [3] introduced an enhanced Faster R-CNN version, Efficient Faster R-CNN, utilizing the EfficientNet-B7 network to accurately detect solder joint defects and PCB components, replacing the original VGG-16 backbone network. Soomro et al. [25] leveraged EfficientNet-B3 to develop a robust PCB recycling classification system. Both experiments underscored a clear correlation between the chosen feature extraction network and detection accuracy. While Faster R-CNN was widely used and excelled in most PCB inspection tasks, a study exploring electronic component detection and localization methods, including transfer learning with Faster R-CNN, unsupervised machine learning clustering (XOR-K-means), and multi-template matching, revealed that combining k-means and CNN classification outperformed Faster R-CNN [26].

B. OTHER NEURAL NETWORK-BASED METHODS

Various studies investigated the effectiveness of different deep neural network architectures for PCB component classification. Lu et al. [27] compared AlexNet and Inception-v3, with Inception-v3 demonstrating superiority in parameters, training speed, and accuracy. In contrast, Wang et al. [28] highlighted AlexNet's exceptional chip defect detection, achieving 99.73% accuracy through specialized methods. Additionally, Reza and Crandall [29] demonstrated the success of IC-ChipNet by adopting a Siamese Network with a ResNet-50 backbone, achieving 83.69% accuracy in IC manufacturer identification, surpassing AlexNet and VGG16.

Autoencoders, employed in unsupervised machine learning, constituted artificial neural networks comprising both an encoder and a decoder. While less explored than mainstream approaches such as Faster R-CNN or YOLO, autoencoder-based methods offered the advantage of learning robust and concise feature representations from input data. De Paulis et al. [30] proposed an advanced PCB inspection system utilizing a skip-connected convolutional autoencoder to identify defect shapes and locations. Makwana et al. [31] introduced PCBSegClassNet, an encoder-decoder architecture crafted for the segmentation and classification of PCB components. The network incorporated a dual-branch design in the backbone to accommodate diverse component sizes and shapes. It also utilized a Texture Enhancement Module (TEM) for refining component boundaries.

III. METHODOLOGY

The methodology involved three primary phases: data preparation, model construction, and model evaluation. YOLOv4 was selected for further refinement based on its successful application in PCB defect detection, as demonstrated by Caliskan and Gurkan [17] and Xin et al. [18]. The proposed solution aimed to improve YOLOv4 by replacing its original CSPDarknet-53 backbone with EfficientNet, a proven and effective architecture for detecting PCB solder joint defects and components [3]. The success of EfficientNet extended beyond that of the PCB industry. For instance, a modified YOLOv4 with EfficientNet-B0 as its backbone was utilized in apple detection, resulting in a lighter model with reduced computational complexity and superior performance compared with YOLOv3 and YOLOv4 [32]. Figure 2 visually illustrates the overall stages of the study.

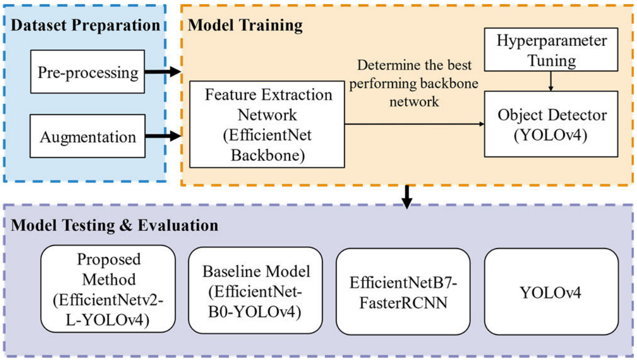

Dataset Preparation
Pre-processing
Augmentation
Model Training
Feature Extraction Network (EfficientNet Backbone)
Determine the best performing backbone network
Hyperparameter Tuning
Object Detector (YOLOv4)
Model Testing & Evaluation
Proposed Method (EfficientNetv2-L-YOLOv4)
Baseline Model (EfficientNet-B0-YOLOv4)
EfficientNetB7-FasterRCNN
YOLOv4

FIGURE 2. Overall stages of the research.

A. NETWORK DESIGN

Object detection architectures typically consists of three main components: backbone, neck, and head. For IC detection on PCBs, the modified EfficientNet-YOLOv4 algorithm was crafted by incorporating EfficientNet as the backbone network, YOLOv4 as the head, and retaining the original neck part of YOLOv4, which included Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PANet) modules.

1) BACKBONE (EfficientNet)

Backbone networks are often derived from classification tasks without a fully connected layer [7]. EfficientNet, introduced by Google in 2019 [33], is one of the current state-of-the-art classification networks. EfficientNet encompassed eight structures, ranging from EfficientNet-B0 to B7, with EfficientNet-B7 having the largest number of parameters. A key merit of EfficientNet lay in its compound scaling method that optimized the width, depth and resolution of the model, resulting in a good trade-off between size and performance. Figure 3 shows the compound scaling of EfficientNet that uniformly scales depth, width, and resolution with a fixed ratio. However, the computational demands of EfficientNet impeded training and inference time, especially for models B6 and B7.

To address those issues, a more compact yet potent iteration, EfficientNetv2, was introduced [34]. EfficientNetv2 included S-, M-, and L-sized structures, with EfficientNetv2-L having the largest size. By incorporating MBConv and Fused-MBConv, EfficientNetv2 integrated Neural Architecture Search (NAS) for optimal block combinations. Its novel non-uniform scaling approach gradually added layers in later stages, enhancing efficiency in training, parameters, and inference speed compared to its predecessor, EfficientNetv1 [34]. EfficientNetv2-M attained comparable accuracy to EfficientNet-B7 with fewer parameters and trained 4.1 times faster [34].

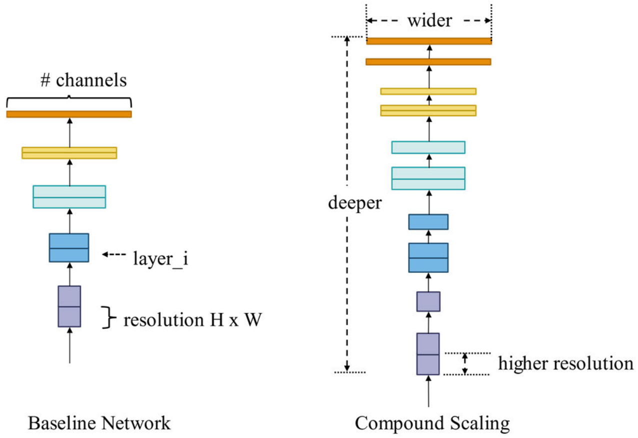


FIGURE 3. Compound scaling method of EfficientNet.

2) NECK

The neck plays a crucial role in aggregating and refining features obtained from the backbone. Its primary function is to enhance the representational power of these features, contributing to more accurate and robust predictions. In the proposed model, the SPP block and modified PANet were retained as the neck, similar to YOLOv4. This design choice ensured continuity with the architecture of YOLOv4. The SPP block is a feature used to capture context at different scales within an image. It uses multiple pooling scales to gather information at various resolutions [35]. PANet introduces a bottom-up pathway on top of FPN to extract and amalgamate additional feature information. Additionally, PANet significantly contributes to refining instance segmentation by preserving spatial data and aiding in accurate pixel localization for mask prediction [36].

3) HEAD (YOLO)

The head component is responsible for generating predictions, encompassing bounding boxes and class scores. YOLOv4 is the fourth version of the YOLO family and represents a mature release that capitalizes on the strengths and insights gained from its earlier versions. Each grid cell in YOLOv4 predicts three bounding boxes with nine anchors based on three different scales and three aspect ratios. These anchors help to determine the actual width and height of the predicted bounding boxes.

In addition, YOLOv4 introduced two techniques known as Bag of Freebies (BoF) and Bag of Specials (BoS) to enhance overall model performance. BoF methods were designed to modify the training strategy or cost without increasing the inference time. BoF included augmentation techniques, such as mosaic data augmentation and Self-Adversarial Training (SAT). Conversely, BoS comprised plugin modules and post-processing methods that significantly improved detection accuracy, albeit resulting in a slight increase in the inference cost [35]. Examples of BoS included Mish activation and the SPP block.

4) EfficientNet-YOLOv4

In the proposed model architecture, the emphasis was placed on enhancing object detection capabilities, particularly for applications like PCB inspection. When designing a detector, prioritizing a higher input network size (resolution) enables effective detection of multiple small-sized objects. Incorporating additional layers expands the receptive field to encompass the augmented input size, while the increased parameters fortify the model's capability to detect diverse object sizes [35]. YOLOv4 embodies these traits, facilitating swift predictions of the object position and classification, making it ideal for real-time applications. As a one-stage detector, YOLOv4 is a state-of-the-art model known for its rapid inference speed. The integration of EfficientNet with YOLOv4 will result in a robust object-detection system. EfficientNet is one of the most potent CNN models, and its updated version, EfficientNetv2-L, exhibits superior parameter efficiency and accuracy.

In this fusion, EfficientNetv2-L replaced YOLOv4's original backbone, thereby enriching the model architecture. The compound scaling method inherent in EfficientNet facilitated the creation of a feature extraction network that is deeper, wider, and higher in resolution, enhancing the model's ability to capture intricate features. The integration of the SPP module from YOLOv4 with the EfficientNet backbone was a strategic move to handle objects of varying scales efficiently. This addition enabled the model to capture multiscale information within the network, and the model became more robust in detecting objects in the input image regardless of their size.

This integration of the proposed method not only improved the efficiency and accuracy of the PCB inspection system but also demonstrated its potential to advance computer vision capabilities, particularly in the domain of object detection. Figure 4 provides a visual representation of the proposed model's architecture, highlighting the replacement of YOLOv4's original backbone with EfficientNetv2-L and illustrating the integration of the SPP module.

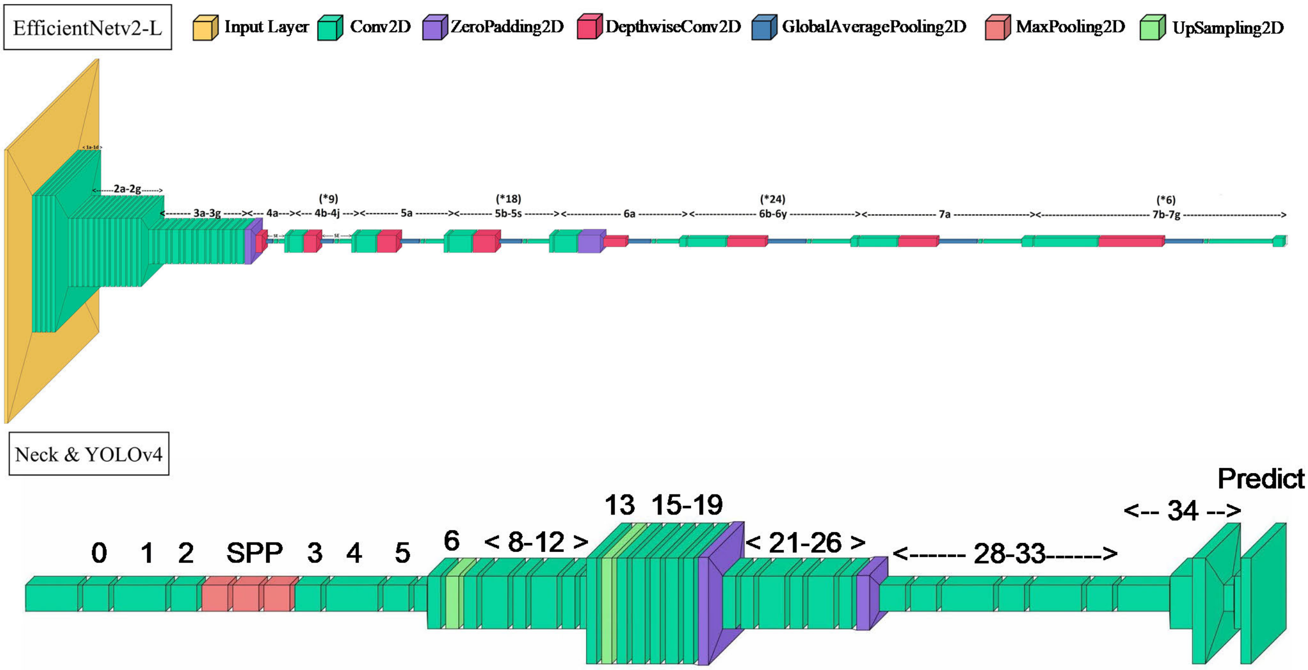


FIGURE 4. Architecture of EfficientNetv2-L-YOLOv4.

The proposed model underwent various enhancements to boost its performance, incorporating Bag of Freebies (BoF) techniques from YOLOv4, adjusting anchor sizes, and refining the loss function. YOLOv4 integrated various BoF techniques into its training pipeline to enhance accuracy. This study specifically explored the impact of experimenting with multiple anchors for a single ground truth and incorporated mosaic data augmentation features. Multiple anchors for a single ground truth are based on the rules in which the intersection over union (IoU) between the anchors and ground truth exceeds a specified threshold. Mosaic data augmentation mixes four training images into one training image, allowing the model to learn different contexts [35].

The YOLOv4 head utilizes anchor boxes to predict the locations and sizes of the objects. However, the dataset used in this study mainly consisted of images containing a single chip object of similar size; k-means clustering for anchor size determination was not directly applicable. In this study, YOLOv4, by default, employed a set of nine anchor sizes for a 416 × 416 image input size: 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326. This set encompassed various scales and aspect ratios and was also known as the anchor size set of YOLOv3, referred to as 'y3' in this study for ease of reference. Additionally, the study explored another set of anchor sizes (13,31, 21,42, 31,15, 34,58, 51,29, 57,98, 78,48, 150,118, 255,323) derived from [2]. This alternative anchor size set was explicitly tailored from a dataset of PCB electronic components. This study referred to this as the 'PCB anchor size set' for ease of reference. Figure 5 shows the distribution diagram of both anchor size sets.

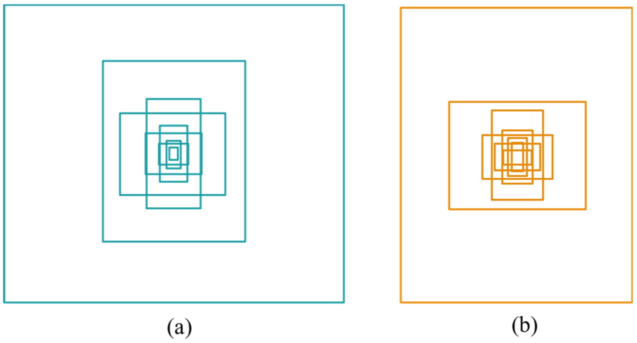


FIGURE 5. 9 anchor size set distribution diagram for (a) y3 anchor size set and (b) PCB anchor size set.

The selection of an appropriate loss function depends on the specific requirements and characteristics of the object detection task. In YOLOv4, the loss function comprises three main components: localization loss (which can include IoU or similar loss), confidence loss, and class loss. Regression loss optimizes various aspects of the predicted bounding boxes, covering localization, confidence scores for object presence, and class predictions. Complete IoU (CIoU), an enhanced version of the IoU metric, is employed as a bounding box regression loss function in YOLOv4 [37]. CIoU loss addresses IoU limitations by considering geometric measures for the complete bounding box information, including overlap area, central point distance, and aspect ratio, providing more precise object localization [37], [38]. Another loss function for bounding box regression is SCYLLA-IoU (SIoU), which focuses on the spatial overlap between bounding boxes and was found to perform better than CIoU [39]. SIoU consists of angle cost, distance cost, shape cost, and IoU costs. Both loss functions were experimented with to determine which was more effective for the proposed model.

TABLE 1. Number of datasets.

Dataset	Number of original images	Number of augmented images	Total
Training	19171	19171	38342
Validation	2197	2197	4394
Testing	5407	5407	10814
Total	26775	26775	53550

TABLE 2. List of augmentations used.

Pixel-level augmentations	Spatial-level augmentations
• HueSaturationValue	• RandomRotate90
• RGBShift	• HorizontalFlip
• ToSepia	• Transpose
• ChannelShuffle	• GridDistortion
• CLAHE	
• RandomBrightness	
• RandomContrast	
• GaussNoise	
• MultiplicativeNoise	
• ISONoise	
• Emboss	
• RandomFog	
• RandomSunFlare	

The formulas of CIoU and SIoU are expressed as follows: (1) and (2).

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{1}$$

where α is a positive trade-off parameter ($\frac{v}{(1-IoU)+v'}$), and v measures the consistency of aspect ratio ($\frac{4}{\pi^2}(artan\frac{w^{gt}}{h^{gt}} - artan\frac{w}{h})^2$) [37], [38].

$$L_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{2}$$

where Δ is distance cost ($\sum_{t=x,y}(1 - e^{-\gamma\rho_t})$), and Ω is shape cost ($\sum_{t=w,h}(1 - e^{-\omega_t})^{\theta}$) [39].

B. DATASETS

The initial dataset consisted of 146 folders with 26,775 images. Within each folder, images were randomly split into a training set (80%) and a testing set (20%). The training set underwent an additional division to create a validation set (10%). The original images in the dataset exhibited a high degree of similarity, prompting the need for increased diversity in features, patterns, or elements across these images. Therefore, the dataset underwent augmentation using the Albumentations library for offline image augmentation. Table 1 provides the dataset breakdown for this study.

Albumentations [40], an open-source Python library compatible with popular deep learning frameworks such as TensorFlow and PyTorch, was utilized for offline image augmentation. The process involved generating new images by applying random transformations to the existing ones. Diverse transformations were applied to each original image to create an augmented version. The randomness of these transformations was determined by the probability assigned to each augmentation. Moreover, specific preconditions were established to determine whether certain transformations were applied. For instance, images deemed dark underwent contrast-limited adaptive histogram equalization (CLAHE) initially to improve brightness. The augmentation methods applied to the original data are summarized in Table 2.

IV. EXPERIMENTS AND DISCUSSION

The software operated on Ubuntu 20.04.2 LTS, utilizing Tesla V100-SXM2-32GB with Driver 470.57.02 and CUDA version 11.4. Python 3.8 served as the primary programming language. The proposed model was configured with specific parameters: 100 epochs, a batch size of 16, input image dimensions of 416 × 416, and an initial learning rate set at 1×10^{-3}, with a 0.5 reduction factor for every 10 epochs using a patience approach. Stochastic Gradient Descent (SGD) was employed as the training optimizer. During inference, the IoU threshold was set to 0.9, and the confidence threshold was 0.8.

The development environment for the proposed model and YOLOv4 utilized Keras-Applications version 1.0.8 with TensorFlow backend version 2.9.1. However, for EfficientNet-B7-FasterRCNN, PyTorch was used, and the batch size was limited to 4 due to GPU memory constraints from its larger model architecture. During the model evaluation phase, consistency among the compared models was maintained by setting identical learning rates and batch sizes. The implementation of the proposed algorithm was based on the repositories of Keras [41] and David [42].

A. BACKBONE COMPARISON

This experiment aimed to identify the best-performing backbone network in YOLOv4. The study compared different backbone networks, with a specific focus on EfficientNet versions 1 and 2. EfficientNetv1 comprises models B0, B1, and B7, ranging from the smallest (B0) to the largest (B7) variant, achieved by scaling the depth of the network. In contrast, EfficientNetv2 offers models S, M, and L, representing small, medium, and large scales based on depth, width, and resolution, respectively. For this comparison, a batch size of 10 was utilized, which was the maximum for EfficientNet-B7-YOLOv4. Table 3 presents the results of a comparative analysis of various EfficientNet backbones integrated into the YOLOv4.

In the first series, EfficientNet-B7 exhibited commendable accuracy with the highest F1-score of 98.43. In contrast, in series 2, EfficientNetv2-L outperformed all others and showcased an impressive F1-score of 98.75 and an mAP of 98.25. It recorded the highest TP count of 10,677 and the lowest FP count of 133, indicating high precision and recall

TABLE 3. Performance comparison of different efficientnet backbones.

Model	EfficientNetv1			EfficientNetv2		
	B0	B1	B7	S	M	L
F1-score	97.45	97.58	98.43	98.18	98.46	**98.75**
mAP@IoU=0.90	96.18	96.20	97.46	97.58	97.81	**98.25**
Precision	97.47	97.61	98.45	98.21	98.47	**98.77**
Recall	97.43	97.55	98.41	98.16	98.44	**98.73**
Inference Time (s)	**0.08**	0.09	0.15	0.10	0.12	0.14
TP	10536	10549	10642	10615	10645	**10677**
FP	274	258	167	194	165	**133**
FN	**4**	7	5	5	**4**	**4**

TABLE 4. Performance comparison of different configurations of the proposed model.

Model	L-ciou-pcban	L-ciou-y3	L-siou-pcban	L-siou-y3	L-siou-y3-BoF	L-siou-pcban-BoF	L-ciou-pcban-BoF	L-ciou-y3-BoF
F1-score	98.23	98.19	98.38	98.30	99.09	99.14	99.16	**99.22**
mAP@IoU=0.90	97.27	97.48	97.72	97.38	98.41	98.66	**98.82**	98.71
Precision	98.25	98.21	98.39	98.33	99.09	99.14	99.16	**99.22**
Recall	98.21	98.16	98.37	98.28	99.09	99.13	99.16	**99.22**
Inference Time (s)	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**
TP	10620	10615	10638	10628	10716	10720	10723	**10730**
FP	189	193	174	181	98	93	91	**84**
FN	5	6	2	5	**0**	1	**0**	**0**
Training Time (days)	4.31	4.31	**3.86**	3.87	3.88	3.89	4.33	4.32

for EfficientNetv2-L. This model surpassed other backbone networks and displayed superior mAP, precision, and recall values. Despite its exceptional performance, EfficientNetv2-L did not have the shortest inference time. EfficientNet-B0 stood out in this regard, requiring only 0.08 seconds per image due to its shallow architecture. In comparison, EfficientNet-B0 processed images approximately 1.73 times faster than EfficientNetv2-L, and EfficientNetv2-L processed images approximately 1.06 times faster than EfficientNet-B7.

B. EXPERIMENT ON ANCHOR SIZE, LOSS FUNCTION AND BOF

The experiments incorporated a combination of diverse loss functions, anchor sizes, and Bag of Freebies (BoF) within the EfficientNetv2-L-YOLOv4 model to evaluate their impact on accuracy. Models were labelled using a specific naming convention: Backbone-lossfunction-anchorsize-BoF, enabling the distinction of various configurations. The training was performed with a batch size of 16, and Table 4 provides details on the outcomes of various configurations of the proposed methods.

- Backbone: EfficientNetv2-L (L)
- Loss function: CIoU (ciou), SIoU (siou)
- Anchor size: YOLOv3 anchor size set (y3), PCB anchor size set (pcban)
- Bag of Freebies: Bag of Freebies (BoF)

When evaluating the proposed model with two anchor size sets, namely the YOLOv3 anchor size set (y3) and the PCB anchor size set (pcban), on the test data, L-ciou-pcban exhibited slightly better performance than L-ciou-y3 in terms of F1-score. The model employing the SIoU loss (L-siou-pcban) also outperformed the model using the y3 anchor size set (L-siou-y3). This experiment suggested that anchor size had an impact on and potentially enhanced the performance of the object detector. In comparing the loss functions, the model that utilized SIoU showed a slightly higher F1-score and TP value than CIoU. Additionally, it was observed from the experiment that the utilization of the CIoU loss function required longer training times compared to the SIoU loss function. This is because SIoU aligns the prediction box more closely with the nearest axis, thus simplifying the regression process and accelerating the training, as stated by [39].

After incorporating the Bag of Freebies (BoF) from YOLOv4 into the proposed models, notable accuracy improvements were observed across all configurations. For instance, L-ciou-y3 achieved a 1.05% accuracy boost upon the inclusion of BoF. However, the combination of BoF,

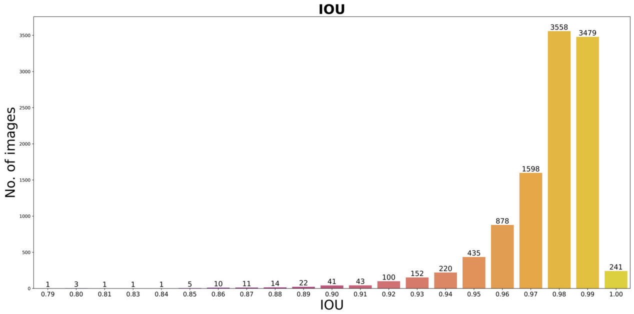


FIGURE 6. IoU graph of EfficientNetv2-L-YOLOv4 (L-ciou-y3-BoF).

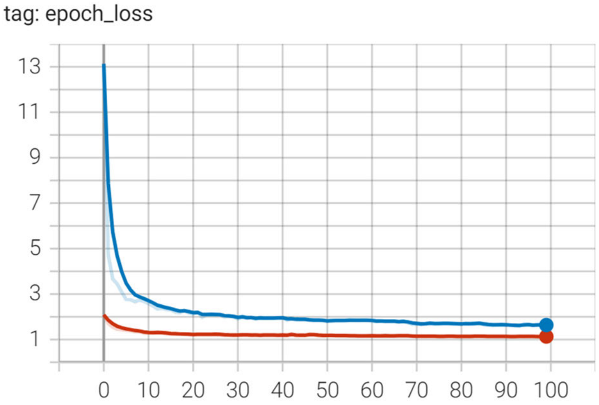


FIGURE 7. Loss graph of EfficientNetv2-L-YOLOv4 (L-ciou-y3-BoF).

SIoU loss function, and pcban anchor size set did not bring significant accuracy advantage to the proposed method. The L-ciou-y3-BoF model, incorporating the CIoU loss function and y3 anchor size set with BoF, stood out with an impressive F1-score of 99.22. Inference time remained consistent across models, at approximately 0.14 seconds.

The improvement in accuracy was attributed to the utilization of multiple anchor points for a single ground truth, enabling the proposed model to select the anchor box that best matched an object's size and aspect ratio, especially for diverse shapes and sizes. When coupled with BoF, the YOLOv3 anchor size set (y3) proved to be better suited for the PCB dataset used in this study. In the IC detection task, the CIoU loss function was applied to address comprehensive bounding box regression errors, proving advantageous in scenarios with varying object sizes and shapes. Consequently, this led to improved localization accuracy compared to using SIoU. Evaluation metrics such as F1-score, mAP, confusion matrix, and inference time exhibited minimal differences among all settings. However, the combination of BoF with the YOLOv3 anchor size set and CIoU loss function in the EfficientNetv2-L-YOLOv4 model (L-ciou-y3-BoF) emerged as the most optimal choice for the IC detection task. Figure 6 and Figure 7 depict the IoU and loss graphs for L-ciou-y3-BoF.

The loss graph illustrated a consistent and gradual reduction in model loss during training, indicating ongoing learning and refinement in predictions. The training loss (blue line) and validation loss (red line) showed a downward trend.

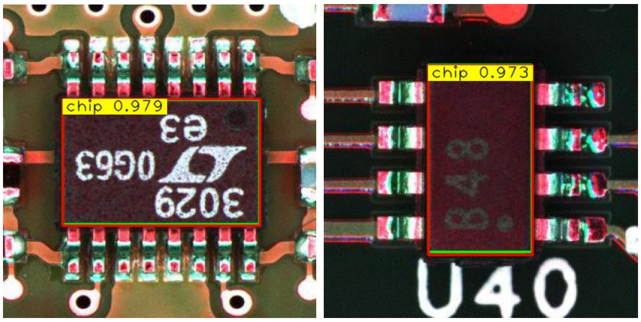


FIGURE 8. Example of predicted results.

The training loss evaluated the model's fit to the training data, while the validation loss measured generalization to the test data. A marginal surpassing of the training loss over the validation loss indicated effective learning without overfitting, highlighting the model's capability to generalize—an essential goal in machine learning.

Below are examples of test results obtained using the proposed method, as illustrated in Figure 8. The predicted IoU scores are visualized, with red bounding boxes representing predictions and green bounding boxes indicating ground truth.

C. TEST THE ROBUSTNESS OF THE MODEL WITH DIFFERENT AUGMENTATION METHODS

The proposed method (L-ciou-y3-BoF) was subjected to various augmentation techniques simulating real-world factors, including blur, size variation, lighting, contrast, color, noise, white spot, and rotation. This evaluation aimed to assess the generalization capabilities of the method in PCB layout scenarios. Despite these simulated environmental changes, the proposed method consistently achieved accurate chip predictions, maintaining IoU scores above 0.90. This high IoU score signified a close alignment between predicted and actual bounding boxes, highlighting the method's robustness. The results showcased the method's reliability in IC inspection tasks and demonstrated its effectiveness even under diverse and challenging image conditions such as noise and varying illumination.

D. PERFORMANCE COMPARISON BETWEEN DIFFERENT MODELS

The enhanced proposed method was compared against the original and other relevant object detection algorithms, including EfficientNet-B0-YOLOv4 [32], EfficientNet-B7-FasterRCNN [3], and the original YOLOv4 [35] to assess its effectiveness. The proposed model, derived from EfficientNet-B0-YOLOv4, acted as the baseline, while EfficientNet-B7-FasterRCNN was drawn from previous PCB component detection studies [3]. This comparison delineated performance distinctions between one-stage (YOLO) and two-stage detectors (Faster R-CNN), both utilizing the EfficientNet backbone but featuring different detector heads. Additionally, the inclusion of YOLOv4 allowed the exploration of potential improvements resulting from backbone

TABLE 5. Model evaluation.

Model	EfficientNet-B0-YOLOv4	EfficientNet-B7-FasterRCNN	YOLOv4	EffcientNetv2-L-YOLOv4
F1-score	97.94	72.77	96.70	**98.96**
mAP@IoU=0.90	96.98	68.26	95.48	**98.23**
Precision	97.96	72.78	96.80	**98.96**
Recall	97.92	72.77	96.60	**98.96**
Inference time (s)	**0.06**	**0.06**	**0.06**	0.10
TP	10589	7869	10446	**10701**
FP	220	2943	345	**112**
FN	5	2	23	**1**

alterations. The models were trained with a batch size of 4 throughout the evaluation due to limited GPU resources to accommodate the large architecture of EfficientNet-B7-FasterRCNN. The hardware and software environments encompassed Ubuntu 20.04.3, utilized an NVIDIA A40 GPU, Driver 495.29.05, and CUDA 11.5. Table 5 displays the results of the model comparison.

Precision and accuracy are significant in real-world applications such as quality control and manufacturing, underscoring the importance of minimizing false positives and ensuring robust IC detection. The EfficientNetv2-L-YOLOv4 model excelled, showcasing impressive performance with an F1-score of 98.96 and mAP of 98.23, demonstrating superior capabilities in accurately identifying ICs. The baseline model, EfficientNet-B0-YOLOv4, achieved commendable accuracy and precision with an F1-score of 97.94 and mAP of 96.98, slightly lower than the former. Replacing EfficientNet-B0 with EfficientNetv2-L resulted in a noteworthy 1.8% accuracy improvement. The proposed EfficientNetv2-L-YOLOv4 marked a significant 2.34% accuracy enhancement over the original YOLOv4 by replacing its CSPDarkNet-53 backbone with EfficientNetv2. EfficientNet-B7-FasterRCNN exhibited the lowest F1-score and mAP among the models.

EfficientNetv2-L-YOLOv4 had a slightly longer inference time, taking 0.102 seconds. In contrast, EfficientNet-B0-YOLOv4 demonstrated faster inference, completing the task in 0.058 seconds. Interestingly, the two-stage EfficientNet-B7-FasterRCNN detector showcased a shorter inference time (0.059 seconds) than the proposed model, while YOLOv4 recorded a similar inference time of 0.061 seconds. The baseline model operated approximately 0.57 times faster than the proposed method.

EfficientNetv2-L-YOLOv4 excelled in accuracy and precision, making it suitable for applications where precision and minimizing false positives were crucial, albeit with slightly longer inference times than other models. Conversely, EfficientNet-B7-FasterRCNN emphasized speed but sacrificed accuracy, while YOLOv4 maintained a balance, albeit with slightly lower accuracy. Two-stage detectors typically leveraged a region proposal step for increased accuracy. However, in this scenario, EfficientNet-B7-FasterRCNN did not notably outperform the one-stage detectors in accuracy. The dataset's characteristics did not fully leverage the two-stage approach, potentially leading to information loss or degradation during transitions between region proposal and object detection, impacting final detection accuracy. While the baseline model outperformed EfficientNet-B7-FasterRCNN in inference speed, EfficientNetv2-L-YOLOv4 lacked this speed advantage, requiring enhancements in inference speed compared to the other three algorithms.

V. CONCLUSION

This research focuses on the challenging task of integrated circuit detection on printed circuit boards by refining the EfficientNet-YOLOv4 algorithm. EfficientNetv2-L-YOLOv4 achieved an impressive F1-score of 99.22 and an inference time of approximately 0.135 seconds through extensive experimentation. Integrating the EfficientNetv2 backbone enhances accuracy beyond baseline models such as EfficientNet-B0-YOLOv4, EfficientNet-B7-FasterRCNN, and the original YOLOv4.

Furthermore, enriching the training dataset with data augmentation techniques improves the proposed model's generalization capabilities. The combination of diverse augmentation methods with EfficientNetv2-L, CIoU loss, YOLOv3 anchor size set (for 416 image size), and Bag of Freebies (L-ciou-y3-BoF) produces optimal outcomes for IC detection. Overall, this study underscores the enhanced EfficientNet-YOLOv4 algorithm's effectiveness in addressing intricate challenges related to IC detection on PCBs, demonstrating superior performance metrics and robustness in handling real-world complexities.

Future research should prioritize exploring various network architectures to advance object detection models, particularly in IC detection on PCBs. Fine-tuning architectural elements like backbone networks, feature extraction layers, and network connectivity promises to enhance model performance in accuracy, speed, and efficiency. Notably, the recent release of YOLOv7, showcasing a 1.5% higher AP than YOLOv4, suggests a promising avenue for refining inspection methods in the industry [43]. Addressing these recommendations could advance IC detection on PCBs, fostering the development of more accurate, robust, and efficient detection methods for industrial inspection applications.

APPENDIX A
PLOT OF PROPOSED METHOD

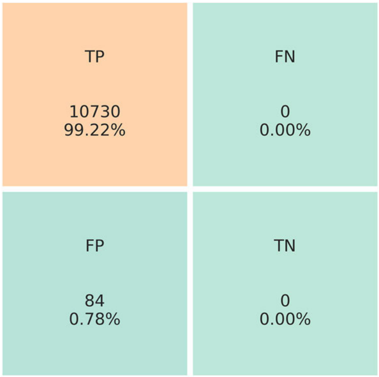


FIGURE 9. Confusion matrix of EfficientNetv2-L-YOLOv4.

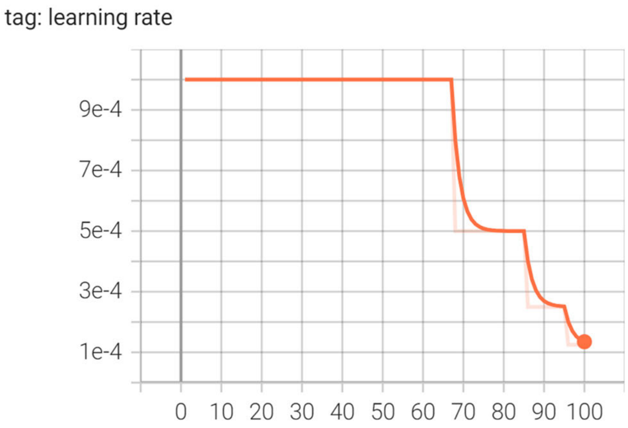


FIGURE 10. Learning rate graph of EfficientNetv2-L-YOLOv4.

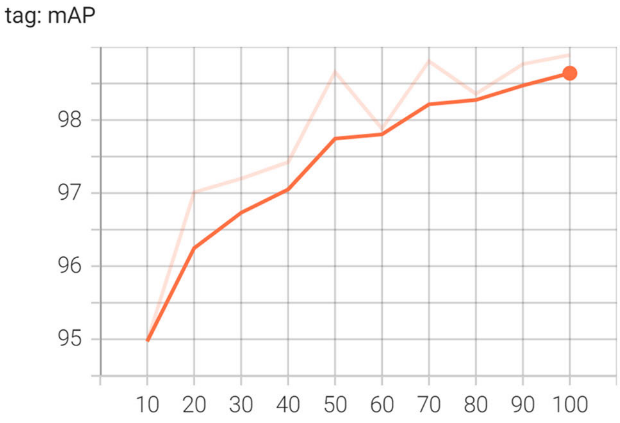


FIGURE 11. mAP graph of EfficientNetv2-L-YOLOv4.

APPENDIX B

TABLE 6. Abbreviations and acronyms.

Acronyms	Definition
AI	Artificial Intelligence
AMI	Acoustic Micro Imaging
AOIs	Automatic Optical Inspection systems
AP	Average Precision
AXI	Automated X-ray Inspection
AutoML	Auto Machine Learning
BoF	Bag of Freebies
BoS	Bag of Specials
BCE	Binary Cross-Entropy
CIoU	Complete Intersection over Union
CLAHE	Contrast-Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Networks
CSP	Cross Stage Partial
ECLAD-Net	Electronic Component Localization and Detection Network
FPN	Feature Pyramid Network
GsIoU	Gaussian Intersection of Union
GIoU	Generalized Intersection over Union
ICs	Integrated Circuits
IoT	Internet of Things
IoU	Intersection over Union
IRT	Infrared Thermography
LB	Loss Boosting
Mask R-CNN	Mask Region-based Convolutional Neural Network
mAP	mean Average Precision
MBConv	Mobile Inverted Bottleneck Residual Blocks
MAM	Multiscale Attention Module
NAS	Neural Architecture Search
PANet	Path Aggregation Network
PCB	Printed Circuit Board
PCBA	Printed Circuit Board Assembly
RCNN	Regions with Convolutional Neural Network
R-FCN	Region-based Full Convolutional Network
RPN	Region Proposal Network
SIoU	SCYLLA-IoU
SAT	Self-Adversarial Training
SPN	Similarity Prediction Network
SPP	Spatial Pyramid Pooling
SGD	Stochastic Gradient Descent
SMT	Surface Mount Technology
SMD	Surface-Mounted Device
TEM	Texture Enhancement Module
VGG	Visual Geometry Group
YOLO	You Only Look Once

ACKNOWLEDGMENT

This work was financially supported by the Collaborative Research in Engineering, Science and Technology (CREST) and SanDisk Storage Malaysia Sdn. Bhd. The authors would like to extend their heartfelt appreciation and deepest thanks to everyone who generously supported and actively contributed to the successful completion of this paper.

REFERENCES

[1] N. Jahani, A. Sepehri, H. R. Vandchali, and E. B. Tirkolaee, ''Application of Industry 4.0 in the procurement processes of supply chains: A systematic literature review,'' *Sustainability*, vol. 13, no. 14, p. 7520, Jul. 2021, doi: 10.3390/su13147520.

[2] J. Li, J. Gu, Z. Huang, and J. Wen, ''Application research of improved YOLO v3 algorithm in PCB electronic component detection,'' *Appl. Sci.*, vol. 9, no. 18, p. 3750, Sep. 2019, doi: 10.3390/app9183750.

[3] F. Fan, B. Wang, G. Zhu, and J. Wu, ''Efficient faster R-CNN: Used in PCB solder joint defects and components detection,'' in *Proc. IEEE 4th Int. Conf. Comput. Commun. Eng. Technol. (CCET)*, Aug. 2021, pp. 1–5, doi: 10.1109/CCET52649.2021.9544356.

[4] N. K. Singh, P. Muthukrishnan, and S. Sanpini, ''System assembly, bring-up and validation,'' in *Industrial System Engineering for Drones: A Guide With Best Practices for Designing*. Berkeley, CA, USA: Apress, 2019, pp. 139–165, doi: 10.1007/978-1-4842-3534-8_5.

[5] S. Cao, I. Parviziomran, H. Yang, S. Park, and D. Won, ''Prediction of component shifts in pick and place process of surface mount technology using support vector regression,'' *Proc. Manuf.*, vol. 39, pp. 210–217, Jan. 2019, doi: 10.1016/j.promfg.2020.01.316.

[6] X. Wu, D. Sahoo, and S. C. H. Hoi, ''Recent advances in deep learning for object detection,'' *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020, doi: 10.1016/j.neucom.2020.01.085.

[7] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, ''A survey of deep learning-based object detection,'' *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

[8] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A. Boulesteix, D. Deng, and M. Lindauer, ''Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges,'' *WIREs Data Mining Knowl. Discovery*, vol. 13, no. 2, p. e1484, Mar. 2023, doi: 10.1002/widm.1484.

[9] W. Pannakkong, K. Thiwa-Anont, K. Singthong, P. Parthanadee, and J. Buddhakulsomsiri, ''Hyperparameter tuning of machine learning algorithms using response surface methodology: A case study of ANN, SVM, and DBN,'' *Math. Problems Eng.*, vol. 2022, pp. 1–17, Jan. 2022, doi: 10.1155/2022/8513719.

[10] M. A. Reza, Z. Chen, and D. J. Crandall, ''Deep neural network-based detection and verification of microelectronic images,'' *J. Hardw. Syst. Secur.*, vol. 4, no. 1, pp. 44–54, Mar. 2020, doi: 10.1007/s41635-019-00088-4.

[11] N. Dave, V. Tambade, B. Pandhare, and S. Saurav, ''PCB defect detection using image processing and embedded system,'' *Int. Res. J. Eng. Technol.*, vol. 3, no. 5, pp. 1897–1901, 2016.

[12] R. Huang, J. Gu, X. Sun, Y. Hou, and S. Uddin, ''A rapid recognition method for electronic components based on the improved YOLO-V3 network,'' *Electronics*, vol. 8, no. 8, p. 825, Jul. 2019, doi: 10.3390/electronics8080825.

[13] T. Khare, V. Bahel, and A. C. Phadke, ''PCB-fire: Automated classification and fault detection in PCB,'' in *Proc. 3rd Int. Conf. Multimedia Process., Commun. Inf. Technol. (MPCIT)*, Dec. 2020, pp. 123–128, doi: 10.1109/MPCIT51588.2020.9350324.

[14] L. H. D. S. Silva, A. A. F. Júnior, G. O. A. Azevedo, S. C. Oliveira, and B. J. T. Fernandes, ''Estimating recycling return of integrated circuits using computer vision on printed circuit boards,'' *Appl. Sci.*, vol. 11, no. 6, p. 2808, Mar. 2021, doi: 10.3390/app11062808.

[15] C. Pramerdorfer and M. Kampel, ''A dataset for computer-vision-based PCB analysis,'' in *Proc. 14th IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, May 2015, pp. 378–381, doi: 10.1109/MVA.2015.7153209.

[16] S.-H. Chen and C.-C. Tsai, ''SMD LED chips defect detection using a YOLOv3-dense model,'' *Adv. Eng. Informat.*, vol. 47, Jan. 2021, Art. no. 101255, doi: 10.1016/j.aei.2021.101255.

[17] A. Caliskan and G. Gurkan, ''Design and realization of an automatic optical inspection system for PCB solder joints,'' in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2021, pp. 1–6, doi: 10.1109/INISTA52262.2021.9548430.

[18] H. Xin, Z. Chen, and B. Wang, ''PCB electronic component defect detection method based on improved YOLOv4 algorithm,'' *J. Phys., Conf. Ser.*, vol. 1827, no. 1, Mar. 2021, Art. no. 012167, doi: 10.1088/1742-6596/1827/1/012167.

[19] C. Guo, X.-L. Lv, Y. Zhang, and M.-L. Zhang, ''Improved YOLOv4-tiny network for real-time electronic component detection,'' *Sci. Rep.*, vol. 11, no. 1, p. 22744, Nov. 2021, doi: 10.1038/s41598-021-02225-y.

[20] X. Liu, J. Hu, H. Wang, Z. Zhang, X. Lu, C. Sheng, S. Song, and J. Nie, ''Gaussian-IoU loss: Better learning for bounding box regression on PCB component detection,'' *Expert Syst. Appl.*, vol. 190, Mar. 2022, Art. no. 116178, doi: 10.1016/j.eswa.2021.116178.

[21] G. Mahalingam, K. M. Gay, and K. Ricanek, ''PCB-METAL: A PCB image dataset for advanced computer vision machine learning component analysis,'' in *Proc. 16th Int. Conf. Mach. Vis. Appl. (MVA)*, May 2019, pp. 1–5, doi: 10.23919/MVA.2019.8757928.

[22] M. A. Mallaiyan Sathiaseelan, O. P. Paradis, S. Taheri, and N. Asadizanjani, ''Why is deep learning challenging for printed circuit board (PCB) component recognition and how can we address it?'' *Cryptography*, vol. 5, no. 1, p. 9, Mar. 2021, doi: 10.3390/cryptography5010009.

[23] C.-W. Kuo, J. D. Ashmore, D. Huggins, and Z. Kira, ''Data-efficient graph embedding learning for PCB component detection,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 551–560, doi: 10.1109/WACV.2019.00064.

[24] L. K. Cheong, S. A. Suandi, and S. Rahman, ''Defects and components recognition in printed circuit boards using convolutional neural network,'' in *Proc. 10th Int. Conf. Robot., Vis., Signal Process. Power Appl., Enabling Res. Innov. Towards Sustainability*. Singapore: Springer, 2019, pp. 75–81, doi: 10.1007/978-981-13-6447-1_10.

[25] I. A. Soomro, A. Ahmad, and R. H. Raza, ''Printed circuit board identification using deep convolutional neural networks to facilitate recycling,'' *Resour., Conservation Recycling*, vol. 177, Feb. 2022, Art. no. 105963, doi: 10.1016/j.resconrec.2021.105963.

[26] C. Yang. (2020). *Machine Learning and Computer Vision for PCB Verification*. KTH KTH Roy. Inst. Technol. Electr. Eng. Comput. Sci. Accessed: Jul. 21, 2022. [Online]. Available: https://kth.diva-portal.org/smash/get/diva2:1529213/FULLTEXT01.pdf

[27] H. Lu, D. Mehta, O. Paradis, N. Asadizanjani, M. Tehranipoor, and D. L. Woodard, ''FICS-PCB: A multi-modal image dataset for automated printed circuit board visual inspection,'' *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 366, Mar. 2020.

[28] J. Wang, X. Zhou, and J. Wu, ''Chip appearance defect recognition based on convolutional neural network,'' *Sensors*, vol. 21, no. 21, p. 7076, Oct. 2021, doi: 10.3390/s21217076.

[29] M. A. Reza and D. J. Crandall, ''IC-ChipNet: Deep embedding learning for fine-grained retrieval, recognition, and verification of microelectronic images,'' in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2020, pp. 1–10, doi: 10.1109/AIPR50011.2020.9425131.

[30] F. de Paulis, R. Cecchetti, C. Olivieri, S. Piersanti, A. Orlandi, and M. Buecker, ''Efficient iterative process based on an improved genetic algorithm for decoupling capacitor placement at board level,'' *Electronics*, vol. 8, no. 11, p. 1219, Oct. 2019, doi: 10.3390/electronics8111219.

[31] D. Makwana, S. C. T. R., and S. Mittal, ''PCBSegClassNet—A lightweight network for segmentation and classification of PCB component,'' *Expert Syst. Appl.*, vol. 225, Sep. 2023, Art. no. 120029, doi: 10.1016/j.eswa.2023.120029.

[32] L. Wu, J. Ma, Y. Zhao, and H. Liu, ''Apple detection in complex scene using the improved YOLOv4 model,'' *Agronomy*, vol. 11, no. 3, p. 476, Mar. 2021, doi: 10.3390/agronomy11030476.

[33] M. Tan and Q. V. Le, ''EfficientNet: Rethinking model scaling for convolutional neural networks,'' in *Proc. 36th Int. Conf. Mach. Learn. (ICML)* May/Jun. 2019, pp. 6105–6114.

[34] M. Tan and Q. V. Le, ''EfficientNetV2: Smaller models and faster training,'' in *Proc. Int. Conf. Mach. Learn.*, vol. 139, Apr. 2021, pp. 10096–10106.

[35] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, ''YOLOv4: Optimal speed and accuracy of object detection,'' 2020, *arXiv:2004.10934*.

[36] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, ''Path aggregation network for instance segmentation,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[37] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, ''Distance-IoU loss: Faster and better learning for bounding box regression,'' in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000, doi: 10.1609/aaai.v34i07.6999.

[38] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, ''Enhancing geometric factors in model learning and inference for object detection and instance segmentation,'' *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8574–8586, Aug. 2022, doi: 10.1109/TCYB.2021.3095305.

[39] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.

[40] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, Feb. 2020, doi: 10.3390/INFO11020125.

[41] Keras-Team. *Keras/Keras/Applications/EfficientNet_v2.py at Master*. Accessed: Aug. 18, 2023. [Online]. Available: https://github.com/keras-team/keras/blob/master/keras/applications/efficientnet_v2.py

[42] David. *Keras-YOLOv3-Model-Set: End-to-End YOLOv4/v3/v2 Object Detection Pipeline, Implemented on tf.keras With Different Technologies*. Accessed: Aug. 18, 2023. [Online]. Available: https://github.com/david8862/keras-YOLOv3-model-set

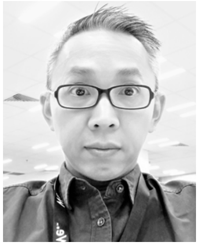
[43] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2023, pp. 7464–7475.

TAY SHIEK CHI received the B.Sc. degree (Hons.) from Universiti Sains Malaysia, Malaysia, in 2020, with an emphasis on multimedia computing, where she is currently pursuing the M.Sc. degree. Her research interests include machine vision and deep learning techniques.

MOHD NADHIR AB WAHAB (Member, IEEE) received the B.Eng. (Hons.) and M.Sc. degrees in mechatronics engineering from Universiti Malaysia Perlis, in 2010 and 2012, respectively, and the Ph.D. degree in robotics and automation systems from the University of Salford, U.K., in 2017. He is currently a Senior Lecturer with the School of Computer Sciences, Universiti Sains Malaysia. His research interests include mobile robotics, computer vision, machine learning, deep learning, artificial intelligence, optimization, navigation, and path planning.

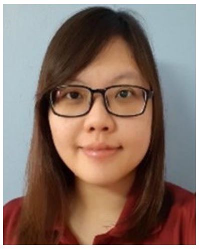
AHMAD SUFRIL AZLAN MOHAMED received the B.I.T. degree (Hons.) from Multimedia University, Malaysia, the M.Sc. degree from The University of Manchester, U.K., and the Ph.D. degree from the University of Salford, U.K. He is currently with the School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia. His research interests include image processing, video tracking, facial recognition, and medical imaging.

MOHD HALIM MOHD NOOR received the B.Eng. (Hons.) and M.Sc. degrees, in 2004 and 2009, respectively, and the Ph.D. degree in computer systems engineering from the University of Auckland, New Zealand, in 2017. He is currently a Senior Lecturer with the School of Computer Sciences, Universiti Sains Malaysia. His research interests include machine learning, deep learning, computer vision, and pervasive computing.



KHAW BENG KANG received the B.E. degree in electronic system engineering from Sheffield Hallam University, in 2000. From 2001 to 2004, he was a Research and Design Engineer with the Renesas Semiconductor (Malaysia) Sdn. Bhd. Since 2005, he has been with Motorola Solutions Malaysia Sdn. Bhd., where he is focusing on two-way radio firmware development. He is currently with Western Digital, Batu Kawan, Penang, as a Specialist in test engineering for solid-state drives.

LIM LAY CHUAN received the bachelor's degree in computer science and engineering from Monash University, in 2000. From 2001 to 2016, he was a Research and Development Engineer in Trek2000 with STEC, Motorola, Malaysia, specializing in embedding programming, NAND storage devices, 2-way radio, and computer bus interfaces. In 2017, he joined Western Digital in test engineering, utilizing various data analytic and machine learning techniques in company's 4th IR revolution and digital transformation. He is the inventor of six patents and three trade secrets. He received the Recognition Award for the "Global Lighthouse Network" when Western Digital Batu Kawan was awarded as the first "Light House" company in Asia by the World Economic Forum (WEF).



LIAU WEI JIE BRIGITTE received the B.Comp.Sc. degree (Hons.) from Universiti Sains Malaysia, Malaysia, in 2020, where she is currently pursuing the M.Sc. degree in computer science. Her research interests include intelligent systems, computer vision, and deep learning techniques.