IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 11 August 2025, accepted 14 September 2025, date of publication 22 September 2025, date of current version 2 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3612382

RESEARCH ARTICLE

Lightweight Multi-Attention Network for SMT Dispensing Electronic Mount Components Identification

SHUCHEN YANG[1], ZHENYI XU[2,3,4], (Member, IEEE), ZHONGHAO WANG[5], YUNBO ZHAO[3,4,6], (Senior Member, IEEE), SHUOQIU GAN[3], SHUMI ZHAO[3], (Member, IEEE), AND JUN HUANG[2], (Member, IEEE)

[1]Jiangsu Engineering Research Center of Key Technology for Intelligent Manufacturing Equipment, Suqian University, Suqian 223800, China
[2]Anhui Engineering Research Center for Intelligent Applications and Security of Industrial Internet, Anhui University of Technology, Ma'anshan, Anhui 243032, China
[3]Anhui Province Key Laboratory of Biomedical Imaging and Intelligent Processing, Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China
[4]Anhui Province Key Laboratory of Intelligent Low-Carbon Information Technology and Equipment, University of Science and Technology of China, Hefei 230027, China
[5]AHU-IAI AI Joint Laboratory, Anhui University, Hefei 230601, China
[6]Suqian Zhongke Intelligent Robot Technology Company Ltd., Suqian 223800, China

Corresponding authors: Zhenyi Xu (xuzhenyi@mail.ustc.edu.cn) and Jun Huang (huangjun.cs@ahut.edu.cn)

This work was supported in part by the Anhui Provincial Natural Science Foundation (2408085MC081 and 2208085UD09), Hefei Municipal Natural Science Foundation (***), open research fund of Anhui Provincial Key Laboratory of Intelligent Low-Carbon Information Technology and Equipment, and the Open Fund of Anhui Province Key Laboratory of Biomedical Imaging and Intelligent Processing (No.24YGXT005), Open Research Fund of Anhui Engineering Research Center for Intelligent Applications and Security of Industrial Internet (IASII25-05).

ABSTRACT During manufacturing process of electronic products, surface mounted technology component dispensing is a key technology in the process of chip production, which affects the quality of products. However, majority dispensing methods utilize rule-based methods, which are not robust to different styles of backgrounds and need to modify parameters when they face different templates. To address the above issues, a YOLOv5-based lightweight multi-attention detection network is proposed for SMT dispensing electronic mount components identification, in which cross and shuffle attention and ghost and multi-attention modules are designed to reduce computational complexity and locate dispensing components rapidly. Moreover, the SMT intelligent dispensing pipeline is realized and experiments on the self-constructed dispensing dataset and experiments show that YOLOv5-Light reaches the best frames 17 FPS on Jetson Nano and satisfied accuracy with mAP@.5 of 99.5% and mAP@.75 of 99.4%. Also, the inference speed is improved from 7.3 ms to 1.0 ms while the space complexity is improved from 10.2 MB to 6.2 MB, which indicates that the proposed dispensing system could implement fast detection under minimizing accuracy loss.

INDEX TERMS Surface mounted technology, dispensing component detection, lightweight attention, edge deployment.

I. INTRODUCTION

The assembly of electronic products is an important link in the manufacturing industry of electronic products. Surface mounted technology (SMT) has become the most popular technology in the electronic assembly industry because of its high reliability and low defect rates of solder joints. The basic elements of SMT includes dispensing, mounting, reflow welding, cleaning, testing and repair. SMT dispensing prevents large components (such as central processing unit (CPU), random access memory (RAM) and peripheral interface (IO)) that have been attached falling off due to the solder paste melting in the secondary furnace. Therefore, it is necessary to use the dispensing machine to place the glue around the component and fix it to the printed circuit board (PCB).

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

Traditional dispensing methods are manual dispensing that is suitable for small-scale production. However, with the rapid development of the electronic industry, manual dispensing is no longer suitable because of the fatigue of employees and it has been replaced by automatic dispensing. The automatic dispensing presets the PCB style and measures in advance the relative position between target components and the mark point on the PCB, then inputs the relative position into the dispensing machine. During dispensing, the dispensing machine identifies the mark point through machine vision, then drive the dispensing manipulator to move to the dispensing position for dispensing. At present, many works contribute to the automatic dispensing.

Sobaszek et al. propose a robot task scheduling mechanism and establish an alternative schedule to control the dispensing [1]. Sunny et al. propose a double-head dispensing machine based on machine vision to improve the accuracy, which can effectively deal with PCB tilting [2]. Murali Ram et al. design a method based on the three-dimensional (3D) pose estimation and 3D reconstruction for 3D object dispensing [3]. Dimitriou et al. use a 3D convolutional neural network to evaluate the dispensing quality [4]. Iftikhar et al. propose a machine model based on the principle of machine learning for SMT [5]. Pagano et al. use point cloud to recognize two-dimensional (2D) and 3D objects and propose a method of coordinate conversion between vision system and the robot [6]. Yongfei et al. propose a template matching method for dispensing system [7]. Peng et al. propose an improved template matching algorithm based on the geometric characteristics of mark points to position the dispensing target [8]. Zhao et al. propose a dispensing machine that could locate the spot position of PCB accurately and complete the dispensing operation through machine vision [9]. Pagano et al. propose a method for trajectory plan of dispensing robots based on image processing and point cloud [10]. Wang et al. propose a method of detecting glue dispensing feature points based on the digital signal processor [11]. Zhao et al. propose a dispensing quality evaluation method based on the combination of entropy weight fuzzy logic and support vector machine [12]. Pei et al. [14] model the energy consumption real-time sensing technology based on mutual inductance and a multi-granularity production line energy consumption. Zhu et al. [15] design intelligent manufacturing experiments based on machine learning for teaching. Chen et al. [16] evaluate an advanced solder paste dispenser and alternative solder paste models to get engineering ready for miniaturization. Thielen et al. [17] investigate a data-driven approach to predict quality of these solder depots in terms of height, area and volume including both process data and previous dispensing quality. In the above methods, major detection methods are based on template matching and image processing to obtain the position of dispensing components. Despite high accuracy could be achieved while it's complex and sensitive to noise. Moreover, if different styles of PCB are given, parameters have to be reset, which is time-consuming and lack of intelligence.

To solve the above challenges, in this paper, we rethink existed detection networks and design an intelligent dispensing system based on lightweight networks, which has smaller parameters and faster inference speed. The contributions of this paper are as follows:

- A lightweight dispensing system is designed, which acquires dispensing component contours by easily-deployed networks and implements motion planning via robot operating system (ROS).
- YOLOv5-Light is proposed for real-time detection and deployment costs. Lightweight cross shuffle attention (CSA) and ghost and multi-attention (GMA) modules are designed to extract features without massive parameters and compact high-level features to avoid meaningless representations.
- Experiments on the dispensing dataset and public benchmark demonstrate that the dispensing system can achieve great precision and speed as well as robustness, which could serve as a strong baseline for SMT dispensing.

II. RELATED WORK

A. LIGHTWEIGHT OBJECT DETECTION

Due to the industrial demand for inference speed, there are many works for improving the real-time performance of models For example, Chen et al. [18] propose a lightweight object detection network in UAV vision based on YOLOv4. Zhou et al. [19] present a dual-path network with a lightweight attention scheme for real-time object detection. Yue et al. [20] develop a lightweight object detection network for single-class multi-deformation objects to promote the practical application of object detection networks. Guo et al. [21] propose a underwater target detection method that optimizes YOLOv8s to make it more suitable for real-time and underwater environments. Hua et al. [22] propose. a new lightweight network for efficient UAV object detection. However, mainstream methods overlook the model inference ability in situations where computing resources are limited, which is difficult to be applied into industrial production such as SMT. Unlike them, for fast production, we rethink the structure of YOLOv5 and empower it with stronger real-time and lightweight, which means it could be deployed into the embedded device to optimize deployment costs.

B. ATTENTION MECHANISM

Although the scaling law suggests that the size of the model helps improve performance on related tasks, effective attention mechanisms can assist the model to learn relevant representations faster. For example, Shen et al. [23] propose a multiple attention mechanism enhanced YOLOX to detect tiny objects against complex backgrounds. Qi et al. [24] introduce an improved YOLOv5 incorporating FasterNet and attention mechanisms to enhance the detection of foreign objects on railways and Airport runways. Zhao et al. [25] propose YOLOv8-QR based on YOLOv8 to detect quick response code defects. Wu et al. [26] propose an improved YOLOv5s network integrating a multi-scale feature fusion module with attention mechanism for crowded road object detection tasks. Ouyang et al. [27] propose an improved underwater object detection model, integrating advanced attention mechanisms and a dilated large-kernel. YOLOv5-Light also benefits from attention mechanisms. But unlike traditional attention mechanisms, which rely on larger networks for feature extraction and global information aggregation, the proposed CSA and GMA modules prioritize computational efficiency and real-time performance. CSA combines MobileNetv3 and ShuffleNetv2 for lightweight feature extraction, while GMA utilizes GhostConv for efficient feature fusion, enabling high-speed detection without sacrificing accuracy.

III. METHODOLOGY

In this section, the dispensing system would be illustrated, which combines object detection and control algorithms to dispense.

A. OVERVIEW

The whole system is divided into two parts: vision and control, which is shown in Fig. 1. In the visual part, it first acquires RGB images of the PCB as well as depth images captured by D455, then the RGB image is fed into YOLOv5-Light to locate dispensing components. The detection results are fed into robust Canny and Hough transform for edge detection to obtain contours of dispensing components. In the control part, the eye-to-hand system is built, and Hough transform results and depth information are combined to carry out coordinate transformation, which means coordinates in the base coordinate system of the manipulator are obtained and then transmitted to ROS for motion planning to control robot manipulation.

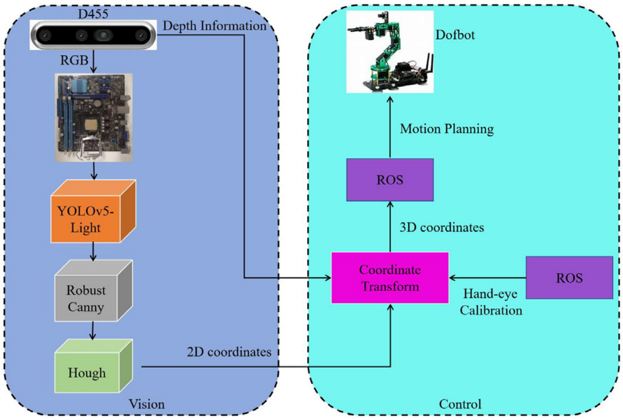


FIGURE 1. The pipeline of the dispensing system, which is divided into visual and control parts. The former acquires contours of dispensing components while the latter is used to motion planning via ROS.

B. YOLOv5-LIGHT

The first step during dispensing requires information where dispensing components exist. Traditional template matching fails to deal with different styles of PCB and has to spend time adjusting parameters. Thus, for lightweight and speed, YOLOv5-Light is proposed to detect dispensing components, which is shown in Fig. 2. Inspired by YOLOv5, YOLOv5-Light consists of the backbone, neck and prediction head. In the backbone, CSA is designed to extract features from RGB images with fewer parameters. In the neck, GMA is designed to pay attention to key expressions and acquire more compact fusion features. Finally, fusion features are fed into the prediction head to predict the category and position of components.

To be specific, given a RGB image $I \in \mathbb{R}^{h \times w \times 3}$, where h and w mean the height and width of the image. it's fed into CSA for extracting features efficiently and $F^{\{i\}}_{CSA} = \phi(I; \theta_{CSA}), i = 1, 3, 5$ would be acquired, where ϕ and θ_{CSA} mean CSA and learnable parameters in CSA. Then, $F^{\{5\}}_{CSA}$ is fed into the convolutional layer and spatial pyramid pooling (SPP) to get the output of the backbone, i.e., $F_b = SPP(Conv(F^{\{5\}}_{CSA}))$, where $Conv$ means the convolutional layer and F_b means the output of the backbone. Next, $F^{\{1,3\}}_{CSA}$ and F_b are fed into the neck to carry out feature fusion and extract key expressions. In the neck, GMA is proposed in top-to-down and down-to-top to emphasize important information without massive parameters, which can be summarized as: $F^i_n = \delta((F_b, F^{\{1,3\}}_{CSA}); \theta_n), i = 1, 2, 3$, where F^i_n means fusion features from different levels. δ and θ_n represent the neck and learnable parameters in δ. Finally, F^i_n is fed into the prediction head to predict the category and position of targets, which can be written as, $(o, x, y, h, w, c) = H(F^i_n; \theta_h)$, where (o, x, y, h, w, c) represents the confidence of the category o, the center point of the prediction box (x, y), the height and width of the prediction box (h, w) and the classification score c respectively. H represents the prediction head and θ_h represents parameters that can be learned in H.

1) CSA

Typical YOLO-series works tend to utilize large-scale networks such as C3 with large parameters to extract features from input images, which will affect the reasoning speed of the model. In order to improve the detection speed while making sure detection accuracy, CSA is designed in the backbone, in which two lightweight networks (MobileNetv3 [43] and ShuffleNetv2 [44]) are considered to acquire lightweight features. To be specific, MobileNetv3 with depthwise convolution is chosen to extract features from the input RGB image I preliminarily. Then, ShuffleNetv2 is introduced after MobileNetv3 to further extract features, in which channel shuffle would improve the efficiency of feature extraction compared to C3. Furthermore, to prevent gradient explosion and preserve the detailed information of I, the basic structure of ShuffleNetv2 is kept but its activation function is replaced with SiLU and improved ShuffleNetv2 is named as Si-ShuffleNetv2, which is shown in Fig. 3. Then, Si-ShuffleNetv2 is repeated by n times to make the best use of features while ensuring computational efficiency.

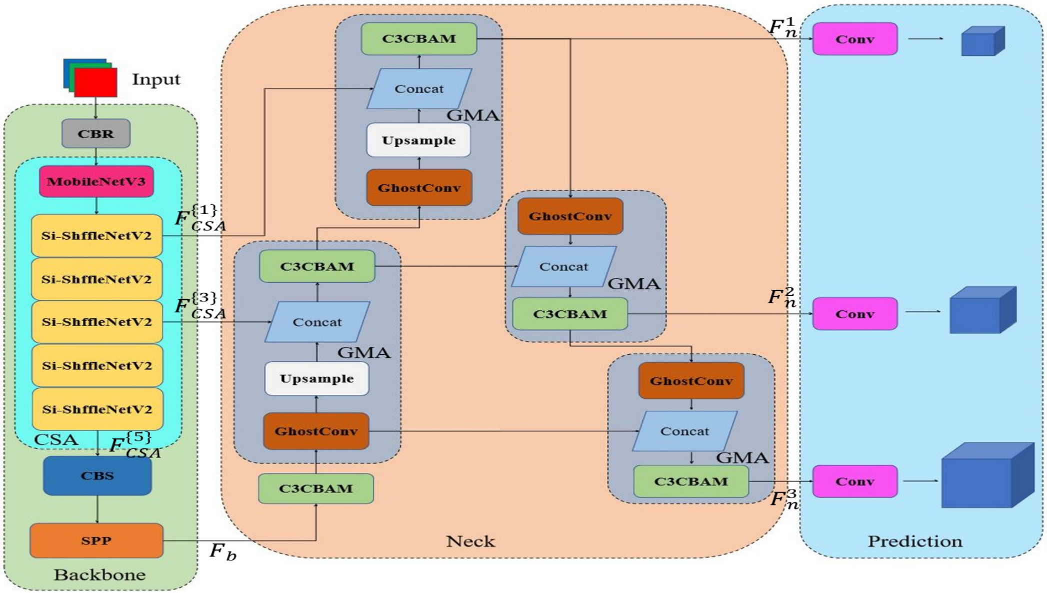


FIGURE 2. The structure of the proposed YOLOv5-Light. Compared with original YOLOv5, CSA is proposed to extract features in a lightweight but effective manner. GMA is introduced into neck to fuse feature maps efficiently.

To utilize features sufficiently without increasing complexity, here n equals 5. Finally, C3 in the original backbone is replaced with the proposed CSA as the improved backbone.

Formally, for the input $I \in \mathbb{R}^{h\times w\times 3}$, it would be fed into MobileNetv3 and $F_M = \epsilon(I)$ could be acquired, where ϵ represents MobileNetv3. Then, F_M is fed into Si-ShuffleNetv2 for further feature extraction and the output of CSA from different levels of Si-ShuffleNetv2 could be obtained, which can be written as,

$$F_{CSA}^{\{i\}} = \lambda_i(F_{CSA}^{\{i-1\}}), \quad i = 1, 2, 3, 4, 5, \tag{1}$$

where λ_i represents the i-th Si-ShuffleNetv2 and $F_{CSA}^{\{0\}} = F_M$.

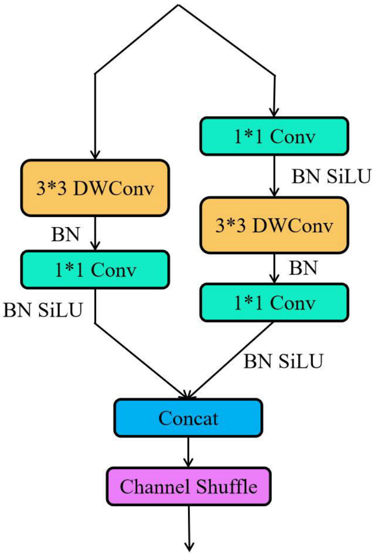


FIGURE 3. The structure of Si-ShuffleNetv2. DWConv represents depthwise convolution. In Si-ShuffleNetv2, the stride is set to 3 to ensure the receptive field and computational efficiency.

2) GMA

CSA is designed for complexity and computation cost in the backbone. Compared with backbone, there are also convolutional layers in the original neck for feature fusion to have a satisfied performance during computation, which takes a lot of memory and reasoning time. Thus, we rethink the structure of the original neck and improve validity of semantics. Following path Aggregation Network (PANet) [46] and feature pyramid network (FPN) [47], GMA includes top-to-down and down-to-top, which is shown in Fig. 4. For top-to-down, GhostConv [45] is utilized to deal with input features at first and then F_{ghost} from GhostConv coulod be obtained, which is computation-efficient without affecting the performance of the model. Next, F_{up} can be acquired through nearest neighbor upsampling F_{ghost}. Then, F_{up} and features from CSA are concatenated to fuse features and get F_{cat}.

In the traditional YOLOv5, high-level fused features will be obtained through C3 from F_{cat}. However, it's expected that the neck could focus on more important semantics while preserve the accuracy of fused features Thus, C3 is considered to focus on more meaningful expressions. As is shown in Fig. 5, CBAM [48] is further introduced into C3. Max pooling and average pooling make C3 extract more compact features and the improved C3 is known as C3CBAM. The whole process of up-to-down can be summarized by:

$$F_{ghost} = GhostConv(F_{input}), \tag{2}$$
$$F_{up} = Upsample(F_{ghost}), \tag{3}$$
$$F_{cat} = Cat(F_{up}, F_{CSA}), \tag{4}$$
$$F_{out} = C3CBAM(F_{cat}). \tag{5}$$

The structure of down-to-top is similar to top-to-down, in which nearest neighbor upsampling is removed and the

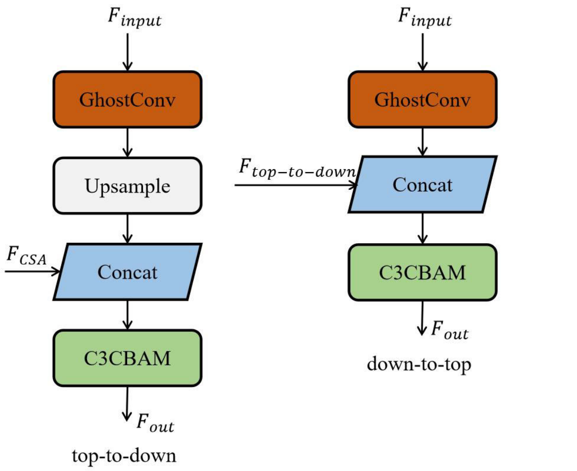


FIGURE 4. The structure of GMA. The proposed GMA is divided into top-to-down and down-to-top. Each relies on lightweight GhostConv to extract effective features and owns the ability to focus on important information through C3CBAM.

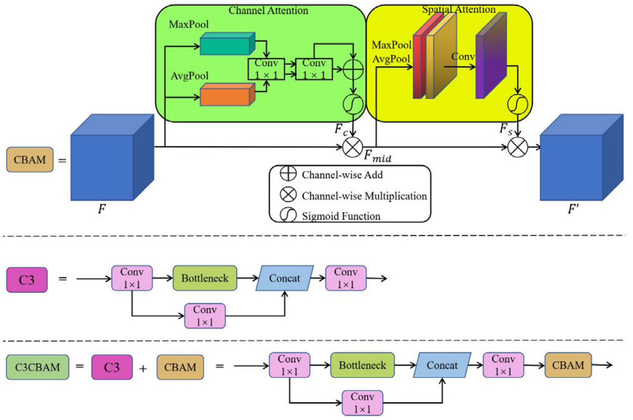


FIGURE 5. The structure of C3CBAM. CBAM is introduced into the output of traditional C3 to enable C3 to extract important features from channels and space.

remaining structure of top-to-down is kept to realize down-to-top. Specifically, GhostConv is used to fusion features at first. Then, features from GhostConv and top-to-down are concatenated and finally C3CBAM is used to extract important features and increase receptive fields.

3) LOSS FUNCTION

The total loss of YOLOv5-Light includes classification loss ($Loss_{cls}$), localization loss ($Loss_{local}$) and confidence loss ($Loss_{conf}$), which is expressed as:

$$Loss_{total} = \alpha Loss_{cls} + \beta Loss_{local} + \gamma Loss_{conf}, \qquad (6)$$

where α, β and γ mean weights of $Loss_{cls}$, $Loss_{local}$ and $Loss_{conf}$ respectively. Binary cross entropy loss (BCELoss) is chosen as classification loss and confidence loss. BCELoss can be calculated as:

$$Loss_{BCE} = -\sum_{n=1}^{N} y_i^* log(y_i) + (1 - y_i^*)log(1 - y_i), \qquad (7)$$

where N is the number of positive samples, y_i is prediction probability and y_i^* is the groundtruth label.

For $Loss_{local}$, CIoU [49] is chosen as the localization loss, which can be denoted as:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(d, d^{GT})}{c^2} + \delta v, \qquad (8)$$

$$IoU = \frac{|D \cap D^{GT}|}{|D \cup D^{GT}|}, \qquad (9)$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{GT}}{h^{GT}} - arctan\frac{w}{h})^2, \qquad (10)$$

$$\delta = \frac{v}{1 - IoU + v}, \qquad (11)$$

where D is the predicted bounding box and D^{GT} is the groundtruth bounding box. $\rho(\cdot, \cdot)$ means the Euclidean distance. d and d^{GT} are the central points of D and D^{GT}. v is a coefficient to measure the consistency of the aspect ratio. w and h represent the width and height of D while w^{GT} and h^{GT} represent the width and height of D^{GT}.

C. EDGE DETECTION

After obtaining positions where components exist, it is necessary to obtain the coordinates of its feature points to carry out dispensing. Thus, feature points are computed through edge detection. To reserve enough memory for YOLOv5-Light on the embedded hardware, parameter-free Canny [51] is chosen instead deep networks such as Mask R-CNN [53] for edge detection. However, in the scene of industrial manufacturing, there's usually interference during SMT dispensing, which will affect Canny. Thus, some changes are introduced to Canny for robustness.

Original Canny includes image filtering, gradient calculation, non-maximum suppression and double threshold setting [51]. Considering that the dispensing system may appear Gaussian noise due to aging equipments and image quality may be affected by light, the median filter is introduced after image filtering, which is defined as,

$$G_2(x, y) = Med(g(x, y)) \qquad (12)$$

where $G_2(x, y)$ is the gray value of pixels after filtering. $Med()$ means the median filter and $g(x, y)$ is the gray value of pixels before filtering.

D. MOTION PLANNING

In the visual part, 2D coordinates of the dispensed components contours are obtained. Then, the internal parameter of the deep camera is used to get the component position (x_C, y_C, z_C) in the camera coordinate system. Finally, 3D coordinates (x_W, y_W, z_W) in the base coordinate system can be calculated as,

$$\begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_W \\ y_W \\ z_W \\ 1 \end{bmatrix}, \qquad (13)$$

where $\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$ is the hand-eye matrix.

To control the robotic arm, it is necessary to acquire the position and pose of the target point. The pose can be set to the fixed according to the glue head vertically downward during dispensing. Therefore, the position and pose of the target point are acquired for motion planning of the robotic arm. In this paper, RRT-Connect [56] is chosen to implement the motion control of the robotic arm.

IV. EXPERIMENTS

In this part, the proposed dispensing system will be tested on the dispensing dataset and employed on Jetson Nano. The effectiveness of YOLOv5-Light would be shown from qualitative and quantitative results.

A. IMPLEMENT DETAILS

Experiments in this paper are based on 64-bit Ubuntu 18.04 and Python 3.7. YOLOv5-Light is implemented under PyTorch. The CPU is an Intel Core i7-11700K. YOLOv5-Light is trained on a NVIDIA GeForce RTX 3090 with the memory of 24GB. During edge deployment, Jetson Nano and Realsense D455 are used to test the ability of real-time. The CPU of Jetson Nano is ARMv8 Processor rev 1 while the GPU of Jetson Nano is NVIDIA Tegra X1 with 4 GB. During training, the learning rate is set to 0.01 and batch size is set to 16. The total epochs are 200 and momentum is set to 0.937. The weight decay is set to 0.0005.

B. DATASET

Considering that there are few publicly available dispensing datasets, the dispensing dataset is made. The images for the dataset come from Kaggle. There are three categories in the dataset, which includes CPU, RAM, and IO, in which there are 8730 CPU instances, 2718 RAM instances, and 12246 IO instances. For annotations, we take the minimum bounding rectangle of each target as the ground-truth bounding box and label them with LabelImg. During the pre-processing stage, reasonable data augmentation is carried out for simulating the complex industrial environment to the dataset. Specifically, changing the brightness of the images as well as adding Gaussian noise [42] are adopted. Moreover, during training YOLOv5-Light, mosaic augmentation is also adopted to rearranging four input images into one single image to improve the ability of robustness of proposed YOLOv5-Light. Then, hue, saturation and value of images are also changed. Finally, scaling operation is used to resize the image to 640 × 640. The hyperparameters of data augmentation are summarized in Table 1. The augmented dispensing dataset is divided at a ratio of 8:1:1. There are 10142 samples in the training set, 1253 samples in the test set and 1127 samples in the validation set. Each picture contains at least one category and some examples are shown in Fig. 6, where the green box means CPU, the purple box means RAM and the red box means IO.

Moreover, K-means clustering is used on the dispensing training set bounding boxes to explore reasonable anchor scales for tiny targets. K-means with distance metric can be written as,

TABLE 1. Hyperparameters for data augmentation.

Parameter	Value
Brightness	0.7
Gaussian noise	0.06
Hue	0.015
Saturation	0.7
Value	0.4
Mosaic	1.0

$$d(box, cluster) = 1 - IoU(box, cluster), \tag{14}$$

where IoU means intersection over union. d means the distance between boxes and clusters. In Fig. 7, it's seen that if K-means doesn't work, YOLOv5-Light will converge slower than using K-means, which indicates that K-means assists the model to find the appropriate scale for tiny objects and make it converge faster.

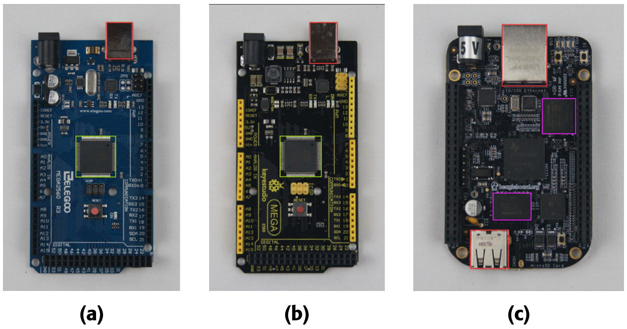


FIGURE 6. Examples from the dispensing dataset. CPU is described in green boxes. RAM is described in purple boxes while IO is described in red boxes.

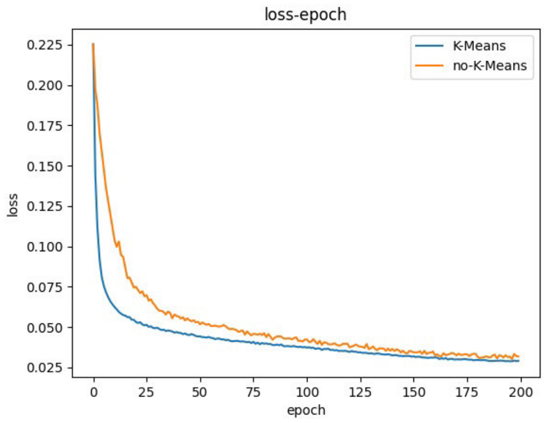


FIGURE 7. The loss-epoch curves with K-means.

C. EVALUATION METRICS

YOLOv5-Light is evaluated from accuracy, the ability of real-time and space complexity. For accuracy, mean average precision (mAP) is chosen as the metric, which can be denoted as:

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}, \tag{15}$$

TABLE 2. Results of different methods on the dispensing dataset.

Model	mAP@.5	mAP@.75	mAP	F1-score	Speed	Params	FPS on Jetson
YOLOv5s	**0.996**	**0.995**	0.966	0.979	7.3ms	10.2MB	3
YOLOv3	0.990	0.990	0.981	0.974	8.7ms	117.8MB	1
YOLOv3-tiny	0.989	0.987	0.944	0.968	1.7ms	17.4MB	7
YOLOv5-Lite-g	0.993	0.977	0.923	0.971	6.8ms	11.3MB	2
YOLOv5-Lite-s	0.991	0.985	0.916	0.971	1.6ms	**3.4MB**	9
YOLOv7	0.971	0.97	0.968	0.947	70.94ms	74.8MB	0.6
YOLOv8	0.987	0.986	0.983	0.970	143.3ms	22.5MB	0.2
Mamba-YOLO	0.983	0.983	0.982	0.950	49.19ms	12.3MB	1
YOLOv6n	0.988	0.988	0.986	0.978	5.2ms	8.7MB	5
YOLOv7-tiny	0.893	0.893	0.791	0.888	4.8ms	12.3MB	6
YOLOv8n	0.985	0.985	0.983	0.976	5.6ms	6.2MB	4
YOLOv9	0.991	0.989	**0.989**	0.975	5.9ms	15.2MB	3
YOLOv10	0.988	0.984	0.984	0.972	5.0ms	5.7MB	6
YOLOv5-Light(Ours)	0.995	0.994	0.975	**0.984**	**1.0ms**	6.2MB	**17**

The best results are in **bold** while the second best are underlined.

where Q is the number of categories. $AP(q)$ means the average precision at a certain category. In this paper, mAP@.5 when IoU is 0.5, mAP@.75 when IoU is 0.75 and average mAP when IoU increases from 0.5 to 0.95 in steps of 0.05 are chosen as evaluation metrics. Moreover, F1-score is chosen as united measurement of precision and recall, which can be expressed as:

$$F1 - score = \frac{2 \cdot P \cdot R}{P + R}, \tag{16}$$

where P represents precision and R represents recall.

For the ability of real-time, frames per second (FPS) testing on Jetson Nano and Real-Sense D455 is chosen as the metric, which can be written as:

$$FPS = \frac{N_{img}}{Seconds}, \tag{17}$$

where N_{img} is the number of processed images and $Seconds$ is the time that reasons these images. Also, the inference speed before deployment is chosen as another real-time metric. For space complexity, the size of the parameters generated after training is chosen. Moreover, FLOPs and the memory usage during the training are also chosen for evaluating complexity.

D. COMPARISON

YOLOv5-Light is compared with different models and relevant results are summarized in Table 2. It can be seen that compared with YOLOv5s, accuracy of YOLOv5-Light is decreased a little but the detection speed increases by 84.1%. Spatial complexity of YOLOv5-Light improves by 39.2% and FPS increases a lot from 3 to 17. Compared with YOLOv3, YOLOv3-tiny, YOLOv4, YOLOv8 and latest Mamba-YOLO, it's found that YOLOv5-Light is excellent in terms of accuracy and speed. Even the fastest YOLOv5-Lite-s, the reasoning speed is still slower than YOLOv5-Light, which improves from 1.6ms to 1.0ms while FPS increases from 79 to 17. Compared with YOLOv5-Lite, YOLOv5-Light still owns higher accuracy and faster speed, but it's more complex than YOLOv5-Lite-s (3.4MB) and less complex than YOLOv5-Lite-g (11.3MB), which indicates low space complexity does not mean fast detection speed. The FPS of YOLOv8 and Mamba-YOLO are 0.2 and 1, which are not advantageous in actual deployment. Moreover, when different models are deployed on Jetson Nano respectively, it's shown that FPS of YOLOv5s is 3 in real-time detection, FPS of YOLOv5-Lite-g and YOLOv5-Lite-s are 2 and 9 respectively while FPS of YOLOv5-Light is 17, which increases a lot and further explains the outstanding performance of lightweight and real-time of YOLOv5-Light.

E. ABLATION STUDY

Next, ablation studies are carried out to analyze effects of proposed modules in YOLOv5-Light and results are shown in Table 3, in which "w/o" in the table means "without" and "w/" means "with". It's seen that the original YOLOv5s holds a higher accuracy while it is time-consuming and complex. When GMA is introduced into YOLOv5s, mAP@.5 drops from 0.996 to 0.99 while inference speed increases by 66.7% and module complexity improves from 10.2MB to 9.4MB. When CSA is introduced into YOLOv5s and the result shows that mAP@.75 decreases from 0.995 to 0.99 but speed improves from 6.3ms to 1.2ms. Further, it's found that compared with GMA, the model performance can be improved better by CSA, which indicates that backbone features extracted by CSA contain important semantic expressions with less computational complexity. Moreover, it also shows that the quality of features extracted by backbone affects detection performance. When SiLU is introduced into CSA, it could achieve higher mAP, which means compared with ReLU, SiLU could retain more valid information. Finally, both of CSA and GMA are introduced into YOLOv5s and the performance reaches the best.

Moreover, YOLOv5-Light has significant advantages in computational efficiency and resource consumption. For example, after introducing CSA and GMA, FLOPs are significantly reduced, in which decreases from 16.4G to 1.6G, and

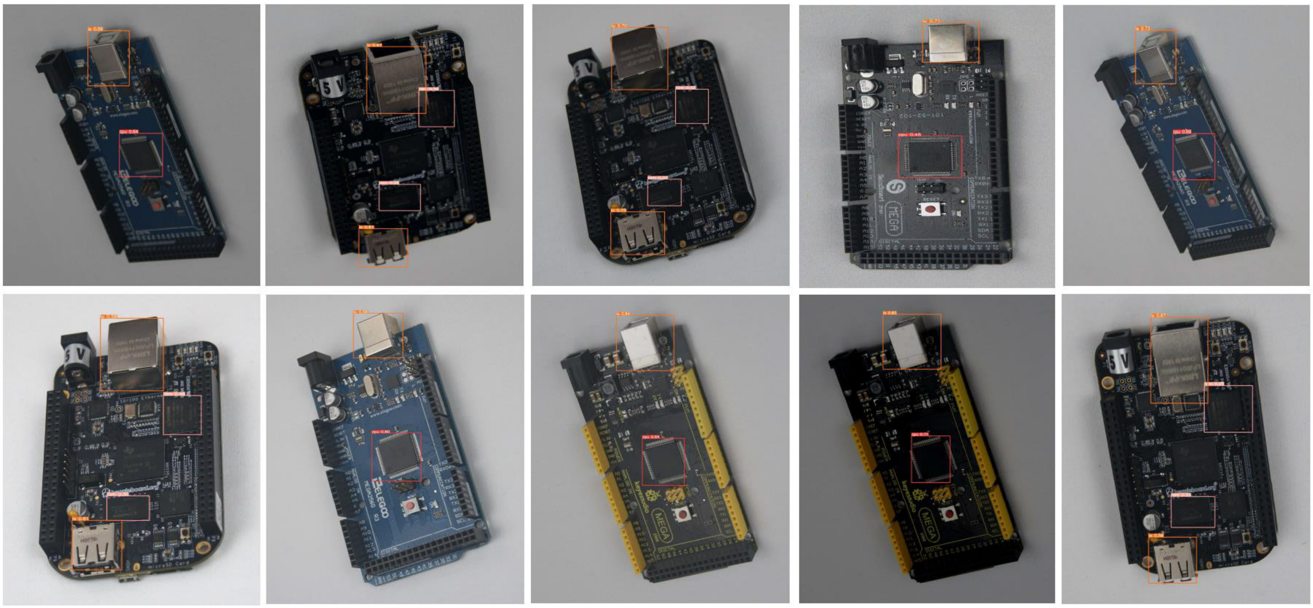


FIGURE 8. The visualization of detection results. YOLOv5-Light is tested in various environments, including angle transformation, brightness transformation and Gaussian noise on different styles of PCB. As PCB images is fed into YOLOv5-Light, the class and position of target will be annotated. In figures, red boxes mean CPU while pink boxes mean RAM and orange boxes mean IO.

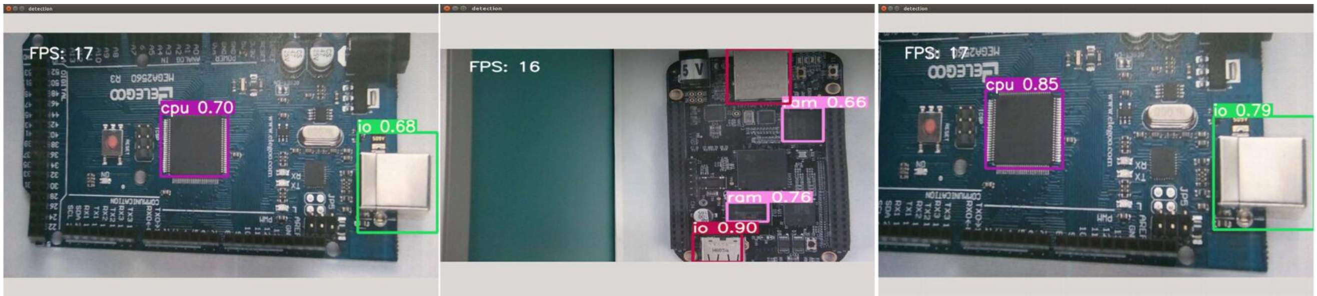


FIGURE 9. Results of real-time detection. YOLOv5-Light is deployed on Jetson Nano and test the performance of real-time with Realsense D455. Results indicate that YOLOv5-Light can quickly detect targets as D455 captures images. In pictures, FPS is displayed in white in the upper left corner. CPU is described in purple boxes. RAM is described in pink boxes while IO is described in green or red boxes.

the memory usage also decreases from 3.59GB to 0.95GB. It's indicated that YOLOv5-Light greatly optimizes the utilization of computing resources while ensuring high detection accuracy. Therefore, YOLOv5-Light not only achieves a balance between accuracy and speed, but also demonstrates significant advantages in computational overhead and memory usage, making it more efficient and scalable in real-time detection scenarios.

F. QUALITATIVE RESULTS

Visualization detection results of YOLOv5-Light are shown in Fig. 8. In the dispensing dataset, there are many different styles of PCB. However, it's seen that YOLOv5-Light could still detect targets regardless of styles of PCB, which is different from traditional template matching and is more intelligent. In addition, even in the case of angle transformation, brightness transformation and Gaussian noise existing, YOLOv5-Light is robust to the noise.

Moreover, for the purpose of lightweight and real-time detection, YOLOv5-Light is deployed on Jetson Nano and realize real-time detection with Realsense D455, which is shown in Fig. 9. It can be seen that YOLOv5-Light could work on dispensing components detection in fast speed. Even if components are partially obstructed, YOLOv5-Light could still detect dispensing targets, which further illustrate the robustness of YOLOv5-Light.

After dispensing components are detected by YOLOv5-Light, edge coordinates will be extracted from the area in which components exist. Note that in order to ensure the effective extraction of contours, predicted boxes are expected to zoom in appropriately during inference and visualization results of three kinds of components are shown in Fig. 10. The first row is the area of the detected components (CPU, RAM and IO) and the second row is the robust Canny for edge detection while the third is the original Canny. It's seen that compared with the improved Canny, original Canny is more susceptible to noise and not suitable for

TABLE 3. Ablation study on different proposed modules.

Model	mAP@.5	mAP@.75	mAP	F1-score	Speed	Params	FLOPs	Memory Usage
YOLOv5s	**0.996**	**0.995**	0.966	0.979	6.3ms	10.2MB	16.4G	3.59GB
YOLOv5s+GMA	0.990	0.990	**0.982**	0.975	2.1ms	9.4MB	13.7G	3.40GB
YOLOv5s+CSA (w/o SiLU)	0.991	0.990	0.949	0.976	1.2ms	6.7MB	2.2G	0.98GB
YOLOv5s+CSA (w/ SiLU)	0.992	0.991	0.952	0.978	1.2ms	6.7MB	2.2G	0.99GB
YOLOv5s+GMA+CSA (w/o SiLU)	0.993	0.992	0.973	0.983	1.0ms	6.2MB	1.6G	**0.93GB**
YOLOv5s+GMA+CSA(Ours)	0.995	0.994	0.975	**0.984**	**1.0ms**	**6.2MB**	**1.6G**	0.95GB

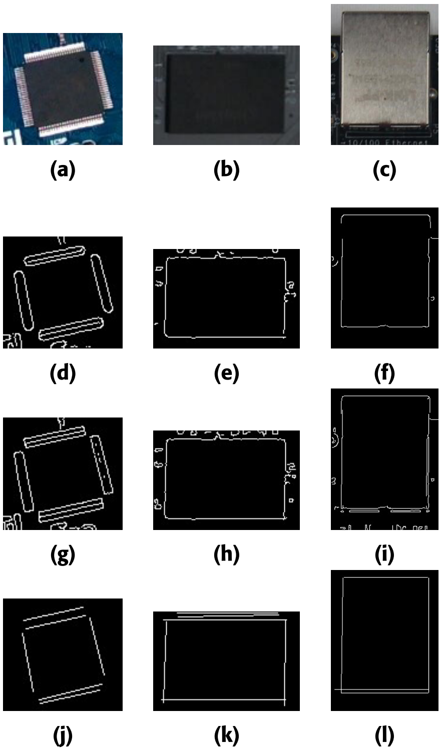


FIGURE 10. The visualization results of three kinds of components. (a) CPU; (b) RAM; (c) IO; (d)~(f) robust Canny; (g)~(i) original Canny; (j)~(l) Hough transform.

the complex and changeable industrial environment. Thus, it shows that the robust Canny could resist noise. The fourth row is visualization of Hough transform detection results. According to the above, it can be seen that components contours could be extracted accurately.

G. GENERALIZATION

To further demonstrate the superiority of YOLOv5-Light, experiments are implemented on the public PCB defect detection dataset (VOC-PCB) [65] for generalization, which are summarized in Table 4. It's seen that even if the dataset is replaced, YOLOv5-Light could still reach satisfied performance in terms of speed and accuracy. To be specific, mAP@.5 of YOLOv5-Light is 0.994, which is the highest among different methods. Also, though mAP@.75 of YOLOv5-Light is not the best, it can be ranked third, which is 0.076 higher than YOLOv5s. For inference speed, it's seen that the proposed YOLOv5-Light could reach the fastest speed on VOC-PCB, which is improved by nearly two orders of magnitude compared to YOLOv8 while owning a smaller number of parameters.

H. HAND-EYE CALIBRATION

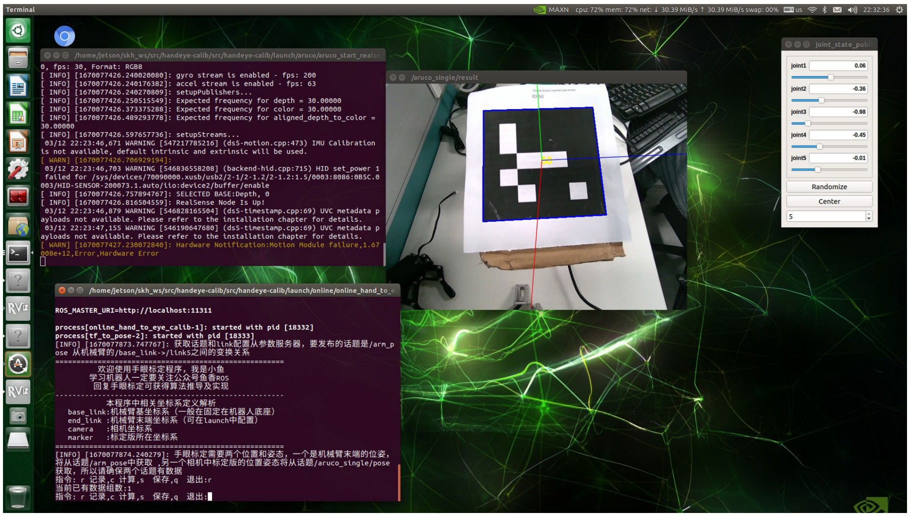
Majority methods need to use software development kit (SDK) of the robotic arm for hand-eye calibration and the precision requirements for the robotic arm are strict. Unlike them, an online calibration method based on ROS is utilized. Specifically, transform (TF) in ROS is used to release the transformation relationship between the base and the end of the robotic arm. Compared with SDK, TF can simply adapt to any kinds of robots in theory, which is independent. Then the hand-eye calibration package in ROS is used to obtain the pose of the calibration board in different movements as shown in Fig. 11. The hand-eye matrix between the camera coordinate system and the robot base coordinate system can be solved by ROS based. Finally, the hand-eye matrix is solved as follows.

$$\begin{bmatrix} -0.998 & 0.050 & -0.023 & -0.018 \\ 0.055 & 0.942 & -0.330 & 0.317 \\ 0.005 & -0.331 & -0.944 & 0.335 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To visualize calibration results, the hand-eye matrix is published to ROS and the result is shown as in Fig. 12. It' seen that the hand-eye matrix could effectively map the relative position of the robotic arm and D455, and the accuracy of the matrix would be reflected by success rates of robotic arm manipulation(Sec. IV-G).

I. DISPENSING ON JETSON NANO

YOLOv5-Light is deployed on Jetson Nano for real-time detection and the test of the whole dispensing system is as follows: after acquiring the pixel coordinates of the contour feature points of the dispensing component through the visual part, the coordinates in the robot base coordinate system are obtained through coordinate transformation. Motion planning is carried out to realize dispensing by ROS, which means Dofbot is controlled to dispense at the edge of components and the error that is shown



```
/home/jetson/skh_ws/src/handeye-calib/src/handeye-calib/launch/aruco/aruco_start_realse
0, fps: 30, Format: RGB8
[ INFO] [1670077426.240020080]: gyro stream is enabled - fps: 200
[ INFO] [1670077426.240176382]: accel stream is enabled - fps: 63
[ INFO] [1670077426.240270809]: setupPublishers...
[ INFO] [1670077426.250515549]: Expected frequency for depth = 30.00000
[ INFO] [1670077426.373375288]: Expected frequency for color = 30.00000
[ INFO] [1670077426.489293778]: Expected frequency for aligned_depth_to_color =
30.00000
[ INFO] [1670077426.597657736]: setupStreams...
 03/12 22:23:46,671 WARNING [547217785216] (ds5-motion.cpp:473) IMU Calibration
is not available, default intrinsic and extrinsic will be used.
[ WARN] [1670077426.706929194]:
 03/12 22:23:46,703 WARNING [546836558208] (backend-hid.cpp:715) HID set_power 1
 failed for /sys/devices/70090000.xusb/usb2/2-1/2-1.2/2-1.2:1.5/0003:8086:0B5C.0
003/HID-SENSOR-200073.1.auto/iio:device2/buffer/enable
[ INFO] [1670077426.757894767]: SELECTED BASE:Depth, 0
[ INFO] [1670077426.816504559]: RealSense Node Is Up!
 03/12 22:23:46,879 WARNING [546828165504] (ds5-timestamp.cpp:69) UVC metadata p
ayloads not available. Please refer to the installation chapter for details.
 03/12 22:23:47,155 WARNING [546190647680] (ds5-timestamp.cpp:69) UVC metadata p
ayloads not available. Please refer to the installation chapter for details.
[ WARN] [1670077427.230072840]: Hardware Notification:Motion Module failure,1.67
008e+12,Error,Hardware Error
```

```
/home/jetson/skh_ws/src/handeye-calib/src/handeye-calib/launch/online/online_hand_to_e

ROS_MASTER_URI=http://localhost:11311

process[online_hand_to_eye_calib-1]: started with pid [18332]
process[tf_to_pose-2]: started with pid [18333]
[INFO] [1670077873.747767]: 获取话题和link配置从参数服务器，要发布的话题是/arm_p
ose 从机械臂的/base_link->/link5之间的变换关系
=======================================================
          欢迎使用手眼标定程序，我是小鱼
        学习机器人一定要关注公众号鱼香ROS
         回复手眼标定可获得算法推导及实现
-------------------------------------------------------
            本程序中相关坐标系定义解析
   base_link:机械臂基坐标系（一般在固定在机器人底座）
   end_link :机械臂末端坐标系（可在launch中配置）
   camera   :相机坐标系
   marker   :标定版所在坐标系
=======================================================
[INFO] [1670077874.240279]: 手眼标定需要两个位置和姿态，一个是机械臂末端的位姿，
将从话题/arm_pose中获取 ,另一个相机中标定版的位置姿态将从话题/aruco_single/pose
获取，所以请确保两个话题有数据
指令: r 记录,c 计算,s  保存,q  退出:r
当前已有数据组数:1
指令: r 记录,c 计算,s  保存,q  退出:
```

joint_state_publ

joint1 0.06

joint2 -0.36

joint3 -0.98

joint4 -0.45

joint5 -0.01

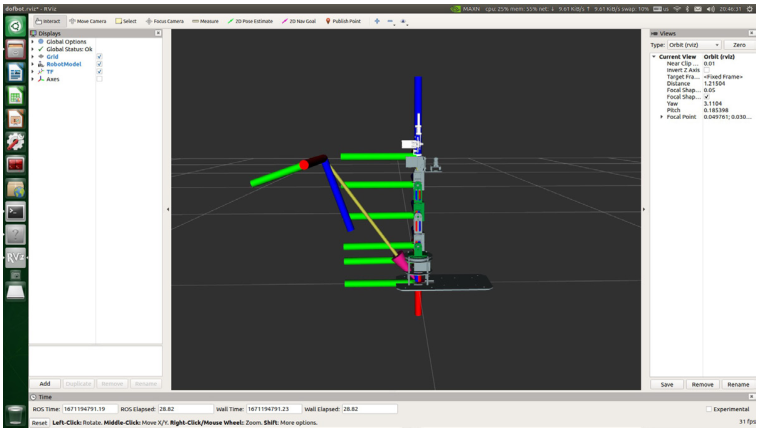
5

Terminal
MAXN cpu: 72% mem: 72% net: ↓ 30.39 MiB/s ↑ 30.39 MiB/s swap: 00% us 22:32:36
/aruco_single/result
Randomize
Center

FIGURE 11. The process of hand-eye calibration.

TABLE 4. Results of different methods on VOC-PCB.

Model	mAP@.5	mAP@.75	mAP	F1-score	Speed	Params
YOLOv5s	0.991	0.811	0.680	0.988	3.3ms	14.4MB
YOLOv3	0.992	0.904	**0.815**	0.986	11.5ms	123.6MB
YOLOv3-tiny	0.989	0.798	0.669	0.985	2.8ms	17.5MB
YOLOv5-Lite-g	0.991	0.748	0.639	0.983	4.5ms	11.3MB
YOLOv5-Lite-s	0.989	0.584	0.564	0.981	4.2ms	**3.4MB**
YOLOv7	0.983	0.565	0.563	0.977	11.6ms	74.9MB
YOLOv8	0.993	**0.934**	0.811	**0.989**	125.3ms	22.5MB
Mamba-YOLO	0.989	0.860	0.728	0.980	11.2ms	12.3MB
YOLOv6n	0.982	0.556	0.545	0.965	2.8ms	8.3MB
YOLOv7-tiny	0.955	0.419	0.492	0.921	3.2ms	12.3MB
YOLOv8n	0.986	0.614	0.575	0.977	2.9ms	6.2MB
YOLOv9	0.986	0.70	0.617	0.984	4.5ms	15.2MB
YOLOv10	0.973	0.612	0.575	0.941	2.6ms	5.7MB
YOLOv5-Light(Ours)	**0.994**	0.887	0.625	0.985	**2.1ms**	8.0MB



Type: Orbit (rviz)

Current View: Orbit (rviz)

Near Clip ...: 0.01

Invert Z Axis: □

Target Fra...: <Fixed Frame>

Distance: 1.21504

Focal Shap...: 0.05

Focal Shap...: ☑

Yaw: 3.1104

Pitch: 0.185398

Focal Point: 0.049761; 0.030...

ROS Time: 1671194791.19 ROS Elapsed: 28.82 Wall Time: 1671194791.23 Wall Elapsed: 28.82

□ Experimental

Reset **Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

31 fps

FIGURE 12. The visualization of hand-eye calibration. The blue line represents the z-axis; the red line represents the x-axis; the green line represents the y-axis. The camera is on the left and Dofbot is on the right.

in Fig.13 is set to ±1cm. The regions between two orange rectangles during dispensing are regarded as successfully operations. The whole process shown in Fig.14, in which the situation of "plan success" means success of motion planning and ROS would send signals to control Dofbot to dispense. In addition, the situation of "plan error" may occur

TABLE 5. Average components dispensing success rates of different models.

Model	CPU	RAM	IO
YOLOv5s	**0.85**	**0.70**	**0.90**
YOLOv3	0.65	0.65	0.85
YOLOv3-tiny	0.65	0.55	0.80
YOLOv4	0.70	0.60	0.80
YOLOv5-Lite-g	0.60	0.55	0.85
YOLOv5-Lite-s	0.55	0.50	0.75
YOLOv8	0.70	0.45	0.85
Mamba-YOLO	0.65	0.45	0.70
YOLOv6n	0.65	0.60	0.75
YOLOv7-tiny	0.45	0.40	0.55
YOLOv8n	0.65	0.55	0.70
YOLOv9	0.70	0.60	0.65
YOLOv10	0.65	0.55	0.65
YOLOv5-Light(Ours)	0.75	0.65	0.8

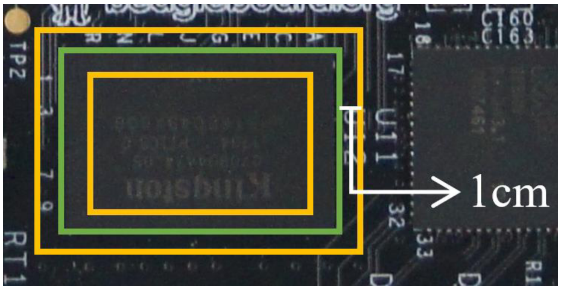
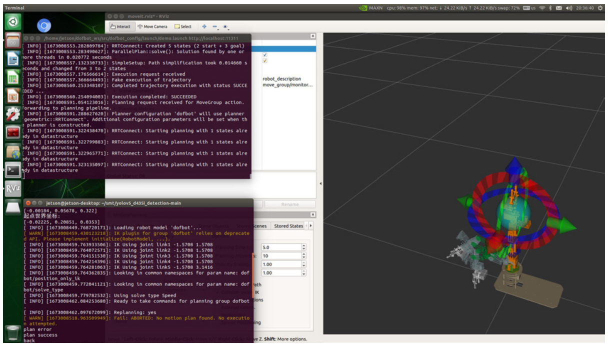


FIGURE 13. The error of dispensing. The green rectangle means the ideal trajectory. The area between two orange rectangles mean acceptable error regions.



```
/home/jetson/dofbot_ws/src/dofbot_config/launch/demo.launch http://localhost:11311
[ INFO] [1673008553.282809784]: RRTConnect: Created 5 states (2 start + 3 goal)
[ INFO] [1673008553.283490627]: ParallelPlan::solve(): Solution found by one or
more threads in 0.020772 seconds
[ INFO] [1673008557.132330733]: SimpleSetup: Path simplification took 0.014660 s
econds and changed from 3 to 2 states
[ INFO] [1673008557.176566614]: Execution request received
[ INFO] [1673008557.366664493]: Fake execution of trajectory
[ INFO] [1673008560.253348107]: Completed trajectory execution with status SUCCE
EDED ...
[ INFO] [1673008560.254094003]: Execution completed: SUCCEEDED
[ INFO] [1673008591.054123016]: Planning request received for MoveGroup action.
Forwarding to planning pipeline.
[ INFO] [1673008591.288627620]: Planner configuration 'dofbot' will use planner
'geometric::RRTConnect'. Additional configuration parameters will be set when th
e planner is constructed.
[ INFO] [1673008591.322438470]: RRTConnect: Starting planning with 1 states alre
ady in datastructure
[ INFO] [1673008591.322799883]: RRTConnect: Starting planning with 1 states alre
ady in datastructure
[ INFO] [1673008591.322965771]: RRTConnect: Starting planning with 1 states alre
ady in datastructure
[ INFO] [1673008591.323135097]: RRTConnect: Starting planning with 1 states alre
ady in datastructure
```

```
jetson@jetson-desktop: ~/smt/yolov5_d435i_detection-main
[-0.00184, 0.05678, 0.322]
起点世界坐标:
[-0.02225, 0.20851, 0.0353]
[ INFO] [1673008449.768720171]: Loading robot model 'dofbot'...
[ WARN] [1673008459.430123218]: IK plugin for group 'dofbot' relies on deprecate
d API. Please implement initialize(RobotModel, ...).
[ INFO] [1673008459.763933506]: IK Using joint link1 -1.5708 1.5708
[ INFO] [1673008459.764072571]: IK Using joint link2 -1.5708 1.5708
[ INFO] [1673008459.764151530]: IK Using joint link3 -1.5708 1.5708
[ INFO] [1673008459.764214396]: IK Using joint link4 -1.5708 1.5708
[ INFO] [1673008459.764281063]: IK Using joint link5 -1.5708 3.1416
[ INFO] [1673008459.764362835]: Looking in common namespaces for param name: dof
bot/position_only_ik
[ INFO] [1673008459.772041121]: Looking in common namespaces for param name: dof
bot/solve_type
[ INFO] [1673008459.779782532]: Using solve type Speed
[ INFO] [1673008462.084253680]: Ready to take commands for planning group dofbot
.
[ INFO] [1673008462.097672099]: Replanning: yes
[ WARN] [1673008518.963509949]: Fail: ABORTED: No motion plan found. No executio
n attempted.
plan error
plan success
back
```

FIGURE 14. The test for the dispensing system. The appearance of "plan success" means the success of motion planning.

when ROS works on motion planning because the accuracy requirements of ROS for the end pose of Dofbot is not within the allowed error range, which may lead to the solution failure. It may caused by parameters, the accuracy of hand-eye calibration or the structure of the robot. Three kinds of dispensing components (CPU, RAM and IO) are tested 20 times on different models respectively and average rates of success are summarized in Table 5.

From Table 2 and Table 5, It's seen that though YOLOv5-Light could not reach the highest success rates, it still achieve the second and own fast inference speed, which means it could realize real-time detection without affecting accuracy a lot. Moreover, it's seen that success rates of CPU and IO are higher than that of RAM, which is because RAM has a similar appearance to CPU but covers an smaller area than CPU, it might fail to detect RAM so that the success rate of RAM is lower than that of CPU during dispensing.

V. CONCLUSION

In this paper, a YOLOv5-based lightweight multi-attention detection network is proposed for SMT dispensing electronic mount components identification, in which cross and shuffle attention (CSA) and ghost and multi-attention (GMA) modules are designed to reduce computational complexity and locate dispensing components rapidly. The proposed model is verified in realistic SMT intelligent dispensing system, which is divided into visual part and control part. Experiments on the dispensing dataset show that the proposed YOLOv5-Light could implement real-time detection without affecting accuracy seriously. The mAP@.5 of YOLOv5-Light on the dispensing dataset is 99.5% and FPS on the NVIDIA hardware Jetson Nano is 17, which means YOLOv5-Light could improve deployment costs and work as a strong baseline in SMT dispensing.

Limitations: However, YOLOv5-Light exhibits notable limitations in practical deployment scenarios. When encountering previously unseen component types or modified PCB layouts, the static nature of its pretrained weights may lead to detection failures or misclassifications. Moreover, successful contour extraction requires near-perpendicular alignment between the PCB plane and camera optical axis. Finally, YOLOv5-Light reaches 17 FPS on Jetson Nano, which still falls below the requirement for high-speed SMT component placement systems and may hinder synchronization with rapid production line operations. We leave these for future works.

REFERENCES

[1] Å. Sobaszek, "A lean robotics approach to the scheduling of robotic adhesive dispensing process," *Adv. Sci. Technol. Res. J.*, vol. 16, no. 5, pp. 136–146, 2022.

[2] S. Agarwal and I. T. W. EAE-Camalot, "Improved process yield with dynamic 'real-time' dual head dispensing," Inst. Printed Circuits APEX EXPO, San Diego, CA, USA, 2020.

[3] S. M. Ram, "Pose estimation and 3D reconstruction for 3D dispensing," Tech. Rep., 2020.

[4] N. Dimitriou, L. Leontaris, T. Vafeiadis, D. Ioannidis, T. Wotherspoon, G. Tinker, and D. Tzovaras, "Fault diagnosis in microelectronics attachment via deep learning analysis of 3-D laser scans," *IEEE Trans. Ind. Electron.*, vol. 67, no. 7, pp. 5748–5757, Jul. 2020.

[5] B. Iftikhar, M. M. Malik, S. Hadi, O. Wajid, M. N. Farooq, M. M. Rehman, and A. K. Hassan, "Cost-effective, reliable, and precise surface mount device (SMD) on PCBs," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 899, no. 1, 2020, Art. no. 012007.

[6] S. Pagano, R. Russo, and S. Savino, "A vision guided robotic system for flexible gluing process in the footwear industry," *Robot. Comput.-Integr. Manuf.*, vol. 65, Oct. 2020, Art. no. 101965.

[7] Z. Yongfei and Z. Tong, "A method of workpiece location based on improved generalized Hough transform," *J. Phys., Conf. Ser.*, vol. 1939, no. 1, May 2021, Art. no. 012079.

[8] G. Peng, C. Xiong, C. Xia, and B. Lin, "A method of vision target localization for dispensing robot based on mark point," *CAAI Trans. Intell. Syst.*, vol. 13, no. 5, pp. 728–733, 2018.

[9] B. Zhao, Q. He, J. Lai, K. Li, C. Zuo, and R. He, "Design of dispensing system based on machine vision," *Int. Core J. Eng.*, vol. 7, no. 7, pp. 125–131, 2021.

[10] S. Pagano, R. Russo, and S. Savino, "A vision guided robot for gluing operations," in *Transactions on Engineering Technologies: World Congress on Engineering*. Singapore: Springer, 2019, pp. 15–28.

[11] N. Wang, J. Liu, S. Wei, and X. Zhang, "A vision location system design of glue dispensing robot," in *Proc. 9th Int. Conf., Intell. Robot. Appl.*, Portsmouth, U.K. Cham, Switzerland: Springer, 2015, pp. 536–551.

[12] L. Zhao, X. Cheng, and Y. Yao, "Online intelligent evaluation of dispensing quality based on entropy weight fuzzy comprehensive evaluation method and machine learning," in *Proc. Int. Conf. Sens., Meas. Data Analytics era Artif. Intell. (ICSMD)*, Oct. 2020, pp. 491–495.

[13] Y. Ting, C.-H. Chen, H.-Y. Feng, and S.-L. Chen, "Glue dispenser route inspection by using computer vision and neural network," *Int. J. Adv. Manuf. Technol.*, vol. 39, nos. 9–10, pp. 905–918, Nov. 2008.

[14] P. Fengque, L. Zhi, D. Wei, M. Song, and H. Song, "Real-time energy consumption sensing system in SMT intelligent workshop," *Mechanics*, vol. 29, no. 5, pp. 387–394, Oct. 2023.

[15] F. Zhu, Z. Chen, W. Zeng, J. Zhang, and S. Li, "Design intelligent manufacturing teaching experiments with machine learning," in *Proc. Int. Conf. Comput. Sci. Educ.* Singapore: Springe, 2023, pp. 232–242.

[16] C.-Y. Chen, Y. Chen, and B. Huang, "Advanced solder dispensing process evaluation for miniaturization and 3D assembly (IMPACT 2024)," in *Proc. 19th Int. Microsyst., Packag., Assem. Circuits Technol. Conf. (IMPACT)*, Oct. 2024, pp. 382–385.

[17] N. Thielen, W. Pan, N. Piechulek, C. Voigt, S. Meier, K. Schmidt, and J. Franke, "Machine learning based quality prediction for solder paste dispensing in electronics production," in *Proc. IEEE 24th Electron. Packag. Technol. Conf. (EPTC)*, Dec. 2022, pp. 858–863.

[18] N. Chen, Y. Li, Z. Yang, Z. Lu, S. Wang, and J. Wang, "LODNU: Lightweight object detection network in UAV vision," *J. Supercomput.*, vol. 79, no. 9, pp. 10117–10138, Jun. 2023.

[19] Q. Zhou, H. Shi, W. Xiang, B. Kang, and L. J. Latecki, "DPNet: Dual-path network for real-time object detection with lightweight attention," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, pp. 4504–4518, Mar. 2025.

[20] X. Yue and L. Meng, "YOLO-SM: A lightweight single-class multi-deformation object detection network," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 3, pp. 2467–2480, Jun. 2024.

[21] A. Guo, K. Sun, and Z. Zhang, "A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection," *J. Real-Time Image Process.*, vol. 21, no. 2, p. 49, Apr. 2024.

[22] W. Hua, Q. Chen, and W. Chen, "A new lightweight network for efficient UAV object detection," *Sci. Rep.*, vol. 14, no. 1, p. 13288, Jun. 2024.

[23] C. Shen, C. Ma, and W. Gao, "Multiple attention mechanism enhanced YOLOX for remote sensing object detection," *Sensors*, vol. 23, no. 3, p. 1261, Jan. 2023.

[24] Z. Qi, D. Ma, J. Xu, A. Xiang, and H. Qu, "Improved YOLOv5 based on the attention mechanism and FasterNet for foreign object detection on railway and airway tracks," in *Proc. Asian Conf. Commun. Netw.*, Oct. 2024, pp. 1–6.

[25] L. Zhao, J. Liu, Y. Ren, C. Lin, J. Liu, Z. Abbas, M. S. Islam, and G. Xiao, "YOLOv8-QR: An improved YOLOv8 model via attention mechanism for object detection of QR code defects," *Comput. Electr. Eng.*, vol. 118, Sep. 2024, Art. no. 109376.

[26] J. Wu, G. Dai, W. Zhou, X. Zhu, and Z. Wang, "Multi-scale feature fusion with attention mechanism for crowded road object detection," *J. Real-Time Image Process.*, vol. 21, no. 2, p. 29, Apr. 2024.

[27] J. Ouyang and Y. Li, "Enhanced underwater object detection via attention mechanism and dilated large-kernel networks," *Vis. Comput.*, vol. 41, no. 11, pp. 8303–8325, Sep. 2025.

[28] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2005, pp. 886–893.

[29] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 1999, pp. 1150–1157.

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

[31] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.

[33] Y. Zhao, H. Huang, Z. Li, H. Yiwang, and M. Lu, "Intelligent garbage classification system based on improve MobileNetV3-Large," *Connection Sci.*, vol. 34, no. 1, pp. 1299–1321, 2022.

[34] J. Zheng, J. Li, Z. Ding, L. Kong, and Q. Chen, "Recognition of expiry data on food packages based on improved DBNet," *Connection Sci.*, vol. 35, no. 1, pp. 1–16, Dec. 2023.

[35] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–16.

[37] B. Dong, Q. Li, J. Wang, W. Huang, P. Dai, and S. Wang, "An end-to-end abnormal fastener detection method based on data synthesis," in *Proc. IEEE 31st Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2019, pp. 149–156.

[38] K. Shi, Z. Xu, Y. Cao, and Y. Kang, "An end-to-end edge computing system for real-time tiny PCB defect detection," in *Proc. Asian Simul. Conf.* Singapore: Springer, 2022, pp. 427–440.

[39] W.-S. Chen, X. Ge, and B. Pan, "A novel general kernel-based non-negative matrix factorisation approach for face recognition," *Connection Sci.*, vol. 34, no. 1, pp. 785–810, Dec. 2022.

[40] Z. Ye, Y. Jing, Q. Wang, P. Li, Z. Liu, M. Yan, Y. Zhang, and D. Gao, "Emotion recognition based on convolutional gated recurrent units with attention," *Connection Sci.*, vol. 35, no. 1, Dec. 2023, Art. no. 2289833.

[41] V. D. Cong and L. D. Hanh, "A review and performance comparison of visual servoing controls," *Int. J. Intell. Robot. Appl.*, vol. 7, no. 1, pp. 65–90, Mar. 2023.

[42] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[43] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[44] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, Apr. 2018, pp. 116–131.

[45] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.

[46] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[47] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[48] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2018, pp. 3–19.

[49] 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.

[50] X. Chen and Z. Gong. *Yolov5-Lite: Lighter, Faster and Easier to Deploy*. Accessed: Sep. 22, 2021.

[51] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Jun. 1986.

[52] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[53] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[54] J. Jiang, X. Luo, Q. Luo, L. Qiao, and M. Li, "An overview of hand-eye calibration," *Int. J. Adv. Manuf. Technol.*, vol. 119, nos. 1–2, pp. 77–97, Mar. 2022.

[55] C. Wu, X. Li, Y. Huang, J. Liu, Z. Qiao, and Q. Hao, "System calibration and pose optimization for robotic-arm-assisted optical coherence tomography," *Proc. SPIE*, vol. 12277, pp. 147–153, Jul. 2021.

[56] J. Chen, Y. Zhao, and X. Xu, "Improved RRT-connect based path planning algorithm for mobile robots," *IEEE Access*, vol. 9, pp. 145988–145999, 2021.

[57] Y. Sun, J. Song, Y. Li, Y. Li, S. Li, and Z. Duan, "IVP-YOLOv5: An intelligent vehicle-pedestrian detection method based on YOLOv5s," *Connection Sci.*, vol. 35, no. 1, Dec. 2023, Art. no. 2168254.

[58] D. Weng, Z. Zhu, Z. Yan, M. Wu, Z. Jiang, and N. Ye, "Lightweight network for insulator fault detection based on improved YOLOv5," *Connection Sci.*, vol. 36, no. 1, Dec. 2024, Art. no. 2284090.

[59] H. Wang, D. Han, M. Cui, and C. Chen, "NAS-YOLOX: A SAR ship detection using neural architecture search and multi-scale attention," *Connection Sci.*, vol. 35, no. 1, pp. 1–32, Dec. 2023.

[60] H. Guijin, W. Ruixuan, X. WuYan, and L. Jun, "Night construction site detection based on ghost-YOLOX," *Connection Sci.*, vol. 36, no. 1, Dec. 2024, Art. no. 2316015.

[61] J. Li, H. Li, X. Zhang, and Q. Shi, "Monocular vision based on the YOLOv7 and coordinate transformation for vehicles precise positioning," *Connection Sci.*, vol. 35, no. 1, Dec. 2023, Art. no. 2166903.

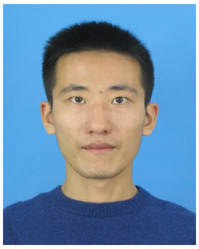
[62] P. Das and S. Chand, "Extracting road maps from high-resolution satellite imagery using refined DSE-LinkNet," *Connection Sci.*, vol. 33, no. 2, pp. 278–295, Apr. 2021.

[63] Z. Wang, C. Li, H. Xu, X. Zhu, and H. Li, "Mamba YOLO: A simple baseline for object detection with state space model," 2024, *arXiv:2406.05835*.

[64] K. Shi, Z. Xu, Y. Cao, L. Zhao, and Y. Kang, "FSPDD: A double-branch attention guided network for few-shot PCB defect detection," *Multimedia Tools Appl.*, vol. 84, no. 19, pp. 1–27, Jul. 2024.

[65] R. Ding, L. Dai, G. Li, and H. Liu, "TDD-Net: A tiny defect detection network for printed circuit boards," *CAAI Trans. Intell. Technol.*, vol. 4, no. 2, pp. 110–116, Jun. 2019.

SHUCHEN YANG received the Ph.D. degree in mechanical engineering from Jilin University, Changchun, China, in 2006. Currently, he is the Dean of the School of Mechanical and Electrical Engineering, Suqian University, a member of the Vocational Education Master's Expert Group of the National Education Professional Degree Graduate Education Guidance Committee, the Director of Jiangsu Key Engineering Research Center for Intelligent Manufacturing Equipment Technology, and a Council Member of Jiangsu Society of Intelligent Manufacturing Engineering.



ZHENYI XU (Member, IEEE) received the Ph.D. degree in control science and engineering from the Department of Automation, University of Science and Technology of China, Hefei, China, in 2020. He is currently an Associate Research Fellow with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center (Anhui Artificial Intelligence Laboratory). His research interests include deep learning, intelligent manufacturing, machine learning, and data mining.

ZHONGHAO WANG received the B.S. degree in printing from Chongqing Jiaotong University, Chongqing, China, in 2024. He is currently pursuing the master's degree in artificial intelligence with the School of Artificial Intelligence, Anhui University. His research interests include deep learning, machine learning, and computer vision.

YUNBO ZHAO (Senior Member, IEEE) received the B.S. degree in mathematics and applied mathematics from Shandong University, in 2003, the M.S. degree from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, in 2007, and the Ph.D. degree from the University of South Wales, U.K., formerly known as the University of Glamorgan, in 2008. He is currently a Professor with the Department of Automation, University of Science and Technology of China. His current research interests include AI-driven human–machine intelligence and AI-driven industrial intelligence.

SHUOQIU GAN received the Ph.D. degree in rehabilitation medicine and physiotherapy from Xi'an Jiaotong University, Xi'an, China, in 2022. She is currently an Associate Research Fellow with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center (Anhui Artificial Intelligence Laboratory). Her research interests include deep learning and machine learning.

SHUMI ZHAO (Member, IEEE) received the B.S. and M.S. degrees from Hefei University of Technology, in 2009 and 2012, respectively, and the Ph.D. degree from the University of Chinese Academy of Sciences, in 2015. From 2018 to 2021, he was a Postdoctoral Researcher with The Hong Kong Polytechnic University. Since 2021, he has been with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center. Currently, he is a Research Fellow with his current research interest focusing on artificial intelligence applications in smart robot and wearable device.

JUN HUANG (Member, IEEE) received the M.S. degree in computer science from Anhui University of Technology, Ma'anshan, China, in 2011, and the Ph.D. degree in computer science from the School of Computer Science and Technology, University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2017, under the supervision of Prof. Qingming Huang. From October 2019 to October 2020, he was a Postdoctoral Researcher with The University of Tokyo, under the supervision of Prof. Kenji Yamanishi.