Sensors and Materials, Vol. 38, No. 8 (2026) 4453–4470
MYU Tokyo

S & M 4576

Enhancing Robotic Arm Control to Support Screwdriver Bit Placement on Assembly Lines

Cheng-Jian Lin,[1] Chi-Huang Shih,[1] and Yeong-Yuh Xu[2*]

[1]Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan
[2]Department of Automatic Control Engineering, Feng Chia University, No. 100, Wenhua Rd., Xitun Dist., Taichung 407102, Taiwan.

(Received January 23, 2025; accepted October 14, 2025)

Keywords: robotic arm, deep learning, attention mechanism, object detection, angle detection

In this study, we built a robotic arm control system that combines image processing and deep learning methods for a practical industrial application to grab screwdriver bits on a running conveyor belt. The target objects, that is, the screwdriver bits, have two appearance characteristics: first, the difference between different types mainly lies in the shape of the head, and the head can be regarded as a small object in the image; in addition, the screwdriver bits are in the shape of an elongated cylinder, so that the angles at which they are distributed on the conveyor belt are also different. Small objects can significantly increase the difficulty of object identification, whereas inaccurate grasping due to angle detection errors may cause damage to the products (i.e., screwdriver bits). Under the requirements of accuracy and reliability, we enhanced the existing vision-based robotic arm control system: (1) in the object detection stage, an attention mechanism is added to improve the recognition performance of small objects, and (2) the principal component analysis (PCA) method is adopted to identify the contour of an object in the presence of image noises. Moreover, we combined edge detection to obtain angle information for the gripper's posture. For these two enhancements, we compared (1) the differences between one-stage object detection methods with and without attention mechanisms and (2) the stability and accuracy of applying PCA to edge detection methods. The experiment data showed that the system proposed in this paper can achieve an object recognition rate of 97.76% and a mean absolute error (*MAE*) of 0 for angle detection. In experiments consistent with industrial application fields, the system obtains successful placement rates of 97% on a static conveyor belt and 87% on a running conveyor belt.

1. Introduction

Classifying and packaging objects still require a lot of human resources at the assembly line. However, the efficiency of human operations is limited. Employees who repeat the packing work for a long time often experience fatigue. In addition to reducing the production speed, it may

*Corresponding author: e-mail: yyxu@fcu.edu.tw
https://doi.org/10.18494/SAM5552

ISSN 0914-4935
https://myukk.org/

lead to operational errors, ultimately degrading the production quality. To address the abovementioned problems, factories have begun to deploy robotic arms on the assembly line to provide a continuous production process without fatigue. Moreover, the robotic arm with a machine vision system can accurately identify and place target objects, thus ensuring good production efficiency.

Traditional machine vision processes include image calibration, recognition, and target positioning. In the image calibration step, images are processed by camera calibration and distortion correction. Then, the image recognition step uses color information to segment the object and further adopts edge detection methods, such as Sobel[1] and Canny,[2] to extract spatial information about the objects. In addition to image processing methods, machine learning techniques are considered in the machine vision system to improve object recognition performance. Ergene and Durdu proposed the grid-based feature extraction method to effectively classify objects on the basis of the bag of words (BoW) model and support vector machine (SVM).[3] However, machine learning requires a lot of computing resources for data training and prediction when feature selection is inappropriate and data are imbalanced.

On the other hand, deep learning can be widely used for object detection based on a two-stage or one-stage method. Fast R-CNN[4] and Faster R-CNN[5] are two common two-stage object detection methods. Ainetter and Fraundorfer employed densely connected feature pyramid networks (FPNs) to extract features and use multiple two-stage detections to achieve grasping posture prediction.[6] Although the two-stage method can improve accuracy, the more regions generated through FPNs, the higher the computational complexity of the region of interest (ROI), resulting in lower recognition speed. Compared with two-stage methods, one-stage methods directly predict the object category and its corresponding bounding box (BBox) only through FPNs. With the reduced computational complexity, one-stage methods can be applied to real-time object detection. Popular one-stage methods include single shot multibox detector (SSD)[7] and you only look once (YOLO),[8–10] which adopt a one-stage method to identify objects moving on a conveyor belt. These works can provide fast and accurate object detection, but may not perform well on multitarget and complex feature classification. To improve the classification performance of a one-stage method, the attention mechanism can focus on key information and ignore unimportant content, enabling the method to better understand and explain input data. Generally, the attention mechanisms are divided into four categories: squeeze-and-excitation (SE) networks,[11] spatial attention model (SAM),[12] channel attention model (CAM),[13] and convolutional block attention module (CBAM).[14] The related works[15–17] combine the attention mechanisms with CNN and YOLO to obtain improved object detection results.

In this study, we developed a vision-based robotic arm control system for the assembly lines of screwdriver bits to perform object placement tasks. A characteristic of screwdriver bits is that the appearances of different types are very similar. They are all slender cylindrical objects with slight differences in head, length, and thickness. In particular, the head can be regarded as a small object in the image system. At the assembly line, screwdriver bits would be randomly placed on the running conveyor belt, and the robotic arm would pick and then place them one by one at specific locations in the storage box for shipment. To realize the automated application

above, we focused on detecting and classifying screwdriver bits on an assembly line using computer vision. While many existing studies focus on general object detection or defect inspection, few have addressed the recognition of small industrial tools, such as screwdriver bits, which often resemble each other closely and require high precision. Additionally, many deep learning models used in industry require powerful computers, which make them difficult to use on edge devices. Our goal is to create a system that can work in real time on an edge device. The developed system employs YOLOv4-CSAM (channel and spatial attention model) to enhance the feature extraction of the target and achieve real-time object detection on running conveyor belts. On the other hand, the success of the screwdriver bit placement task depends on accurately obtaining the spatial information of the slender object, including the center point coordinates and object angle, so that the robotic arm can correctly grab the screwdriver bit and ensure undamaged shipment quality. This system considers the principal component analysis (PCA) technique[18,19] to reduce the dimensionality of the spatial information of the identified object and further performs edge detection to calculate its angle. We conducted data set establishment, system implementation, and experiments at the assembly line site.

The performance analysis of the developed system is divided into technique and system aspects. For the technique aspect, YOLOv4-CSAM outperforms other one-stage methods without an attention mechanism and achieves the highest *mAP* of 97.76%. The PCA-based angle detection method can obtain a more accurate and stable object angle than Canny, achieving a maximum absolute error (*MAE*) of 0. For the system aspect, the recognition rate of objects on the static conveyor belt is 100%, and the successful object placement rate is 96.67%; the object recognition rate on the running conveyor belt is 100%, and the successful placement rate is 86.67%.

The structure of this paper is organized as follows. In Sect. 2, we describe the architecture of the vision-based robotic arm control system, including camera correction and robotic arm coordinate conversion. In Sect. 3, we explain the YOLOv4-CSAM network for object detection and the PCA-based angle detection method. Experiments and performance comparison results are reported in Sect. 4. Finally, this study's conclusions and future works are discussed in Sect. 5.

2. Robotic Arm Grasping System for Screwdriver Bits

2.1 Materials and configuration

In this study, we aim to construct a robotic arm control system that grasps scattered screwdriver bits on a conveyor belt to the specified locations in a storage box. Figure 1 shows a snapshot of a vision-based robotic arm grasping system at the assembly line site. Twelve screwdriver bits of different types are scattered on the conveyor belt. The robotic arm uses a TM robot (type no. TM5-700) with a movable range of 700 mm and a load capacity of 6 kg.[20] The conveyor belt has a speed regulator, LUYANG US425-01, to adjust the object transfer speed finely. The camera uses Logitech C922 with a maximum resolution of 1080p and a maximum frame rate of 30 frames per second (fps). The diagonal field of view (dFoV) is 78° for Logitech

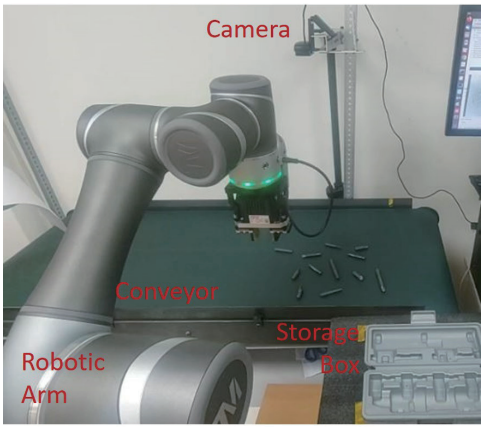

Fig. 1. (Color online) Components and robotic arm grasping system setup.

C922. The robot operation system (ROS) with MoveIt is installed for the robotic arm to conduct the motion planning for robotic arm control.[21] Figure 2 shows twelve screwdriver bits with their respective types under consideration in this paper.

As shown in Fig. 1, the vision-based system adopts the eye-to-hand method to deploy the camera. In the eye-to-hand deployment, the camera is fixed on one side of or above the working area.[22] Therefore, the object's position, angle, and movement in the working area can be observed. Furthermore, the captured images are less disturbed by light, and the robotic arm can be expected to pick up objects more stably and accurately. Compared with the eye-to-hand deployment, the eye-in-hand deployment method installs a camera at the end of the robotic arm, and the object's position can be detected through the movement of the robotic arm.[23] Since the camera is attached to the robotic arm, the detection area changes with the movement of the arm. The eye-in-hand deployment also demands a high-quality camera with depth detection capability to prevent frequent object detection failures.

2.2 System overview

Figure 3 shows the robotic arm control system developed in this study. The system can be divided into four stages: calibration, object and angle detection, coordinate conversion, and gripper control. The calibration stage aims to correct camera distortions and to extract the coordinate relationship between the camera and the target object. The spatial information specified by the coordinate relationship is adopted as an important reference in the coordinate conversion stage, where the object and angle detection results would be converted to the robotic arm's coordinate system. This study's camera calibration process follows the work of Zhang[24] using a 9 × 6 checkerboard. Then, a two-point marking method is used to calculate the coordinate relationship between the camera and the robotic arm.[25] In the two-point marking method, two points are located on the diagonal of an image, and each of the two calibration points constructs a circle with its center. The two-point marking process includes the following: (1) the pixel coordinates of the calibration point (i.e., the circle's center) are first calculated through Hough

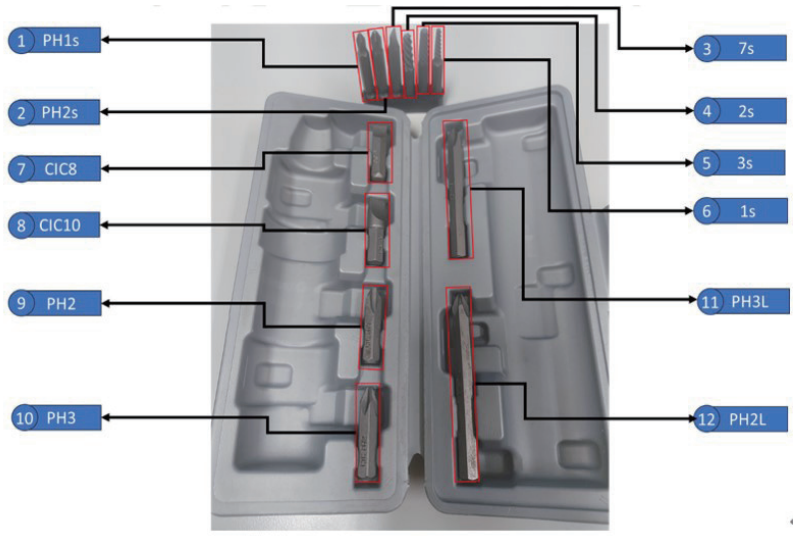

Fig. 2. (Color online) Storage box with twelve screwdriver bits.

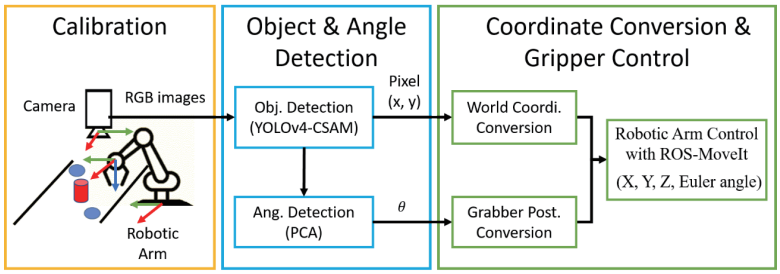

Fig. 3. (Color online) Three-stage vision-based robotic arm grasping system.

circle detection,[26,27] and (2) the front end of the gripper is moved to align the calibration points, and ROS-MoveIt can obtain the world coordinates of two points.[21]

In the object and angle detection stage, the object detection method recognizes the target object with its image coordinates. Generally, a bounding box covering the detected object is available with its image-level coordinates (x, y). However, the angle information is not included in the bounding box output. The angle detection method calculates the object angle (θ) on the basis of the bounding box output to obtain the gripper posture for object grasping. In the final stage, the coordinate conversions from the 2D spatial information $\{x, y, \theta\}$ to control the gripper and robotic arm are available in ROS.

3. Object and Angle Detection

We adopted YOLOv4 as the core model and integrated the attention mechanism to enhance the detection performance for similar small objects in the object detection method. Performance comparisons between the proposed YOLOv4-CSAM and other one-stage methods on

screwdriver bits are discussed in Sect. 4. In addition to the object's position provided by the YOLO-based bounding box, the angle detection method combines PCA and edge detection to extract the essential feature vector, resulting in a robust angle output in the presence of environmental interference such as light.

3.1 Attention-based object detection

Figure 4 presents the architecture of YOLOv4 and its derived enhancement using the attention mechanism. YOLOv4 comprises the following three layers: backbone, neck, and head.[28] The backbone layer is built on CSPDarknet53, whereas CBL, spatial pyramid pooling (SPP), FPN, and path aggregation network (PANet) constitute the neck layer. The three different CBLs in PANet merge to construct the head layer. A CBL with the most detailed spatial information applies the attention mechanism for small-object detection to enhance the feature extraction capability before CBL merging.

General attention mechanisms include channel attention, spatial attention, and the hybrid model (i.e., channel and spatial attention). Channel attention aims to adaptively adjust each channel's importance to capture the image's characteristics better. Specifically, a channel attention mechanism is realized by calculating the weight of each channel and performing a weighted average of the feature map. Given a 2D image with height H and width W, the input feature map of C channels can be denoted as $F \in R^{C\times H\times W}$. As shown in Fig. 5(a), the channel features can be further extracted from F by using average-pooling and max-pooling operations, generating $F^c_{avg} \in R^{C\times 1\times 1}$ and $F^c_{max} \in R^{C\times 1\times 1}$, respectively. These two-channel maps are then

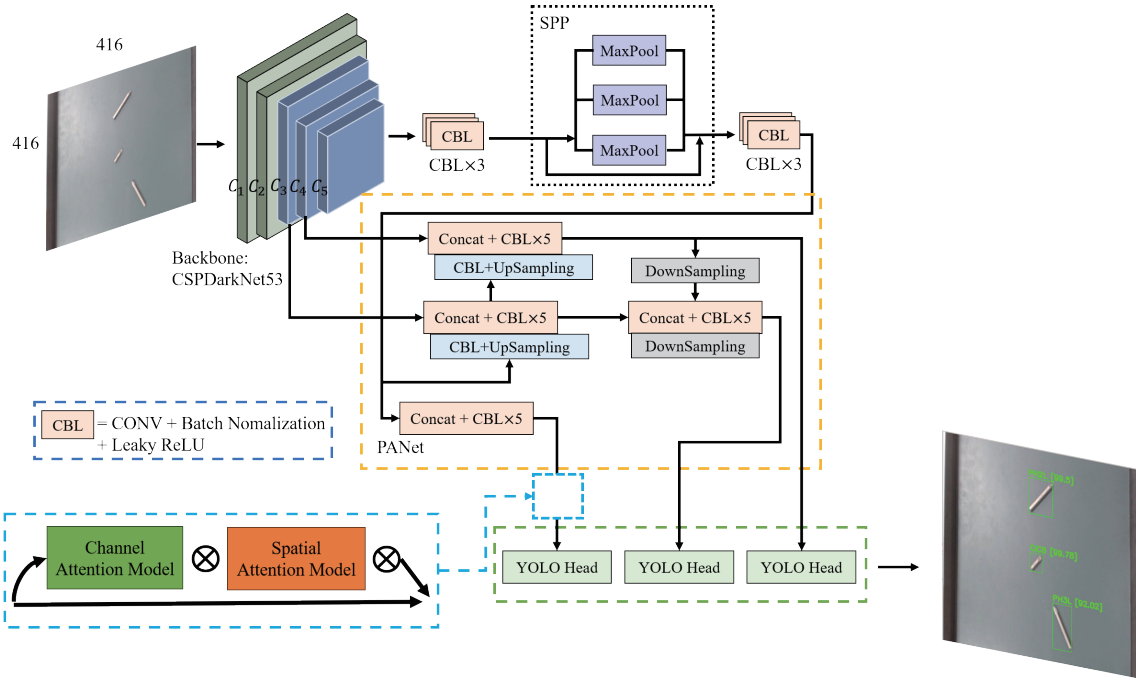

Fig. 4. (Color online) System architecture of YOLOv4-CSAM.

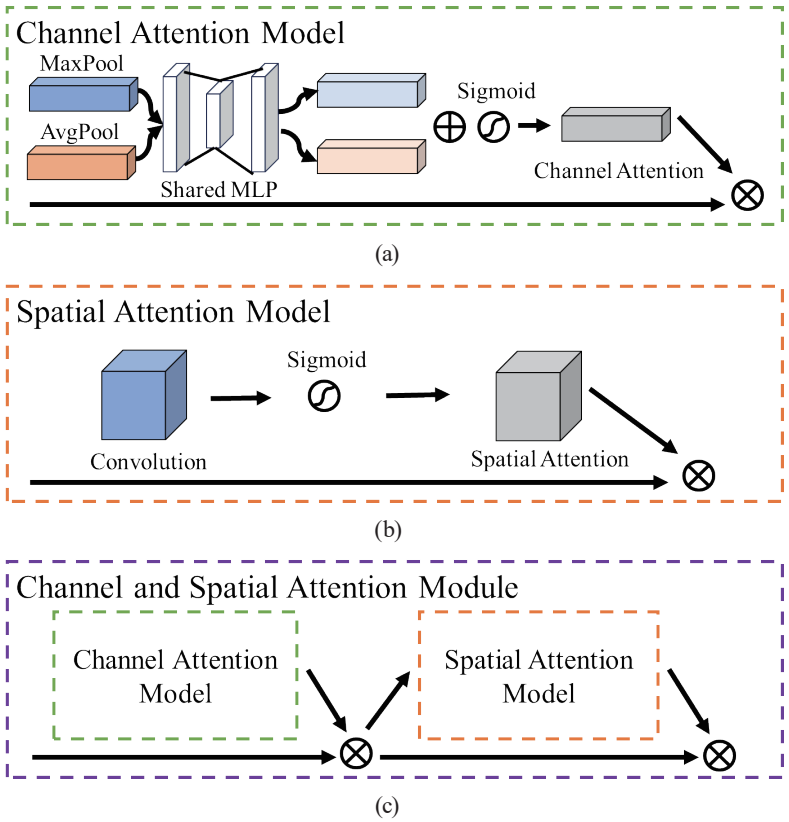

Fig. 5. (Color online) Attention models: (a) channel attention, (b) spatial attention, and (c) channel and spatial attention.

forwarded to a shared multilayer perceptron (MLP) network to obtain convolution results. After the element-wise summation of the convolution results, the sigmoid operation is adopted to activate the channel attention map:

$$M_c(F) = Sigmoid\left(MLP\left(avg(F)\right) + MLP\left(\max(F)\right)\right), \tag{1}$$

where *avg*(.) is the average-pooling function, max(.) the max-pooling function, *MLP*(.) the MLP operation function, *Sigmoid*(.) the activation function using sigmoid operation, and $M_c \in R^{C\times1\times1}$. Finally, the output feature map becomes

$$F_c = M_c(F) \otimes F, \tag{2}$$

where ⊗ stands for element-wise multiplication.

When processing images, spatial attention can focus on areas of interest to improve object detection or classification accuracy. As shown in Fig. 5(b), the spatial attention model uses a standard convolution layer to convolve the input feature map *F*, and finally activates it using sigmoid operation to obtain the spatial attention weight feature map M_s:

$$M_s(F) = Sigmoid\left(Conv^{3x3}(F)\right), \tag{3}$$

where $Conv^{3x3}$ is a 3 × 3 convolution operation, and $M_s \in R^{1\times H\times W}$. In the spatial attention, the output feature map is further calculated as

$$F_s = M_s(F) \otimes F. \tag{4}$$

Hybrid attention combines the characteristics of the above two attention mechanisms, paying attention to spatial location and channel characteristics, and can achieve better object detection results in complex images. In hybrid attention, channel attention and spatial attention are arranged sequentially. Referring to Fig. 5(c), the input feature map F is first processed according to Eq. (2), and the channel feature map Fc is regarded as an input of spatial attention to obtain a hybrid feature map:

$$F_{cs} = M_s(F_c) \otimes F_c. \tag{5}$$

To correctly detect the screwdriver bits with different heads, we considered the channel-spatial attention mechanism (CSAM) to build a YOLOv4-CSAM system. As shown in Fig. 3, CSAM is embedded between the PANet and the head layer in YOLOv4, enabling YOLO to focus on small objects adaptively.

We adopted an attention structure inspired by CSAM owing to its ability to effectively integrate both channel attention and spatial attention in a lightweight and modular form. CSAM provides good balance between accuracy and speed, which makes it suitable for our task of recognizing and locating screwdriver bits in a real-time system that runs on edge devices. It is also easy to integrate with YOLO models without significantly slowing down or complicating the system.

We also examined alternative attention methods, including self-attention,[29] ECA,[30] neighborhood attention,[31] and MILA.[32] However, each has some drawbacks for our use case. For example, ECA is very fast and lightweight, but it only focuses on channel information and does not include spatial attention, which is important for detecting small shape differences in our objects. MILA offers joint attention across spatial and channel dimensions, but its structure is more complex and less compatible with the lightweight, real-time constraints of our system. Self-attention and neighborhood attention offer powerful global modeling capabilities; however, they are computationally intensive and less suitable for deployment on resource-limited edge platforms. Considering the need for high detection accuracy, fast inference, and architectural simplicity, we determined that a CSAM-based attention design provided the most practical and effective solution for our system.

3.2 PCA-based angle detection

In the object detection stage, only the category and center coordinates of the object are available, and the angle and direction information cannot be directly obtained. The side length ratio of the bounding box typically indicates that a slender object is placed vertically (90°) or horizontally (0 or 180°), even though the object is in the tilted state (45 or 120°). Precise angle detection can be beneficial to successfully grasp objects while ensuring that possible damage to the object caused by grasping is minimized.

The angle calculation typically employs an edge detection method to find the object's contour. Then, the placement angle of the object can be computed on the basis of the relation between the contour and the horizontal line. However, Canny cannot detect minimum rectangular edges for all objects in the presence of light interference, resulting in the skewing of the edges, and eventually, significant angular errors can be observed.

To address the abovementioned problem, we employed the PCA method to improve edge detection accuracy. For a BBox covering the detected object, the grayscale conversion is first conducted to reduce unnecessary noise in the BBox image. Then, we adopted PCA as a preprocessing stage before edge detection, and the object angle can be calculated according to the following procedures.

In the grayscale conversion step, the 3D RGB image would be converted into a one-dimensional grayscale image. From the grayscale image, features of the object, including those related to the contour, such as vertices, turning points, or other representative points, can be further extracted. By applying PCA to extracted features, the eigenvector matrix E with the object center point as the origin can be obtained:

$$\begin{bmatrix} v_{1x} & v_{2x} \\ v_{1y} & v_{2y} \end{bmatrix} = E\,, \tag{6}$$

where $\{v_{1x}, v_{1y}\}$ and $\{v_{2x}, v_{2y}\}$ are two feature vectors whose correlation coefficient is zero, and $\{v_{1x}, v_{1y}\}$ has a larger vector than $\{v_{2x}, v_{2y}\}$. The eigenvector represents the direction with the largest standard deviation of gray-scale values, that is, the main shape of the object. Since the screwdriver bits are long strip objects, the vector with a larger value in the feature vectors overlaps with the main direction of the object. Therefore, we can use the Arctan function to obtain the radian represented by the vector and then convert it into an angle θ.

$$\theta = \arctan\left(\frac{v_{1y}}{v_{1x}}\right) \times \frac{180}{\pi} \tag{7}$$

The angle in Eq. (7) is also the z-axis yaw angle of the gripper.

4. Experimental Results

In this study, we collected data and conducted experiments in a real-world assembly line. YOLOv4 and its attention-based enhancement were compared with other one-stage methods for object detection. For angle detection, the performance improvement provided by PCA was investigated at various placement angles. Finally, experiments were conducted to evaluate the system's performance in the static and running conveyor belt scenarios.

The experimental environment used in this study consists of an Intel i7-770 CPU and an NVIDIA GeForce RTX 2060 GPU under Ubuntu 18.04. Detailed system specifications are provided in Table 1.

The experimental models include YOLOv4, YOLOv4-tiny, YOLOv4-CSAM, YOLOv7, YOLOv7-tiny, YOLOv7-CSAM, and SSD. All models were developed and evaluated in Python 3.8. The SSD model was implemented using the PyTorch framework, whereas the YOLO model was implemented using a compiled version of the Darknet framework. Attention mechanisms were incorporated into the YOLO series models. We measured the number of parameters and estimated the inference time for each model. The parameter settings, counts, and estimated latencies for each model are listed in Table 2.

As shown in Table 2, lightweight models such as YOLOv7-tiny and YOLOv4-tiny achieved over 100 FPS, whereas larger models, including YOLOv4 and YOLOv7-CSAM, as well as our final model YOLOv4-CSAM, were able to run at 30–50 FPS. All selected models achieved inference speeds that meet the timing requirements of our robotic system. Since the robotic arm uses a prediction mechanism based on the conveyor and arm speeds, the system can reliably perform real-time grasping.

Table 1
System specifications.

Workstation	
CPU	Intel(R)Core(TM)i7-770@3.60GHz
GPU	NVIDIA GeForce RTX 2060
Memory	16 GB
Operating system	Ubuntu 18.04

Table 2
Model size, training settings, and estimated inference speed (RTX 2060).

Model	SSD	YOLOv7	YOLOv7-tiny	YOLOv7-CSAM	YOLOv4	YOLOv4-tiny	YOLOv4-CSAM
Input size	300 × 300	416 × 416	416 × 416	416 × 416	416 × 416	416 × 416	416 × 416
Backbone network	VGG16	E-ELAN	E-ELAN	E-ELAN	Darknet53	CSP Block	Darknet53
Initial learning rate	0.001	0.00261	0.00261	0.00261	0.001	0.00261	0.001
Momentum	0.9	0.9	0.9	0.9	0.9	0.9	0.9
No. of training iterations	20000	6000	6000	6000	6000	6000	6000
No. of parameters (M)	~24.0	~37.0	~6.2	~38.2	~63.9	~6.0	~65.2
Estimated inference time (ms/image)	~16	~18–22	~7–9	~20–24	~28–32	~7–10	~30–35
Estimated FPS	~62	~45–55	~110–140	~40–50	~30–35	~100–140	~28–33

4.1 Dataset and performance metrics

In this research, we focused on the application domain of screwdriver bit identification and localization in industrial assembly lines. Currently, there is no publicly available benchmark dataset suitable for this specific task, object class, and application scenario. Owing to this limitation, we developed a proprietary dataset containing images of screwdriver bits from various angles for model training and testing.

We captured every frame of objects moving on the conveyor belt from video clips for the twelve different types of screwdriver bits (refer to Fig. 2). The screwdriver bits were placed in random directions. The data set has 4105 images, of which 521 have multiple types coexisting in one image. The ratio of the training set to the validation set is 8:2.

The object detection methods use Precision, Recall, $F1$ score, *mAP*, and *IoU* as performance metrics in the experiment. IoU mainly refers to the joint area at the intersection of the real (i.e., ground-truth) and prediction boxes. The larger the area, the higher the overlap ratio between the prediction and ground-truth boxes, which also means the higher the score. Accordingly, IoU is computed as

$$IoU = \frac{Area_{pre} \cap Area_{gt}}{Area_{pre} \cup Area_{gt}}, \tag{8}$$

where $Area_{pre}$ and $Area_{gt}$ are the areas of the prediction and ground-truth boxes, respectively. The threshold was set as 0.5 in this study; when IoU is larger than 0.5, the object is considered to be correctly predicted. The comparison results between the predicted and real boxes can further derive true positive (*TP*), false positive (*FP*), false negative (*FN*), and true negative (*TN*). On the basis of these four derived parameters, Precision and Recall are given by

$$Precision = \frac{TP}{TP + FP}, \tag{9}$$

$$Recall = \frac{TP}{TP + FN}. \tag{10}$$

The $F1$ score is calculated as the harmonic mean of the Precision and Recall scores.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{11}$$

Average Precision (*AP*) is the area of the Precision–Recall curve. For n categories, *mAP* is the average *AP* of all categories.

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{12}$$

4.2 Object detection results

Table 3 shows the detection results for YOLOv4-CSAM and other compared one-stage methods, namely, YOLOv7,[33] YOLOv7-tiny,[33] YOLOv4, and YOLOv4-tiny.[28] From Table 1, we can observe that (1) YOLO-based methods perform better than SSD; (2) among YOLO-based methods with the same generation, the tiny version has lower values than the original one for all performance metrics; and (3) YOLOv4-CSAM can outperform YOLOv4 and YOLOv7, and YOLOv4 has higher values than YOLOv7 for all metrics.

Experimental results showed that YOLOv4-CSAM achieved the best object detection capabilities in this study, accurately identifying all target categories and significantly outperforming YOLOv7 and YOLOv7-tiny in overall recognition performance. Furthermore, YOLOv4-CSAM further improves its feature extraction and classification accuracy compared with the original YOLOv4 by introducing an attention module. We note that the YOLO family has evolved to YOLOv12, with architectures such as DETR[34] and DiffusionDet.[35] However, these models require high computational resources and are not suitable for deployment on the edge computing platform used in our experiments. Therefore, we chose YOLOv4-CSAM, which demonstrated the most consistent performance and cost-effectiveness, as the final deployment model.

The object detection results can be summarized into two key points: (1) Under the same dataset and training conditions, different object detection methods have significant differences in their recognition rates for screwdriver bits. In particular, newer YOLO systems may not necessarily outperform older YOLO systems. For example, in Table 3, YOLOv4 has higher values than YOLOv7 for all performance metrics. (2) Attention-based methods can help improve the recognition performance of object detection methods for small objects. On the basis of the analysis of the experimental results above, we concluded that for object detection applications in real-world industrial fields, it is important first to select an object detection method that matches the target objects and field constraints (such as computing power). Attention-based methods can also be used to increase recognition efficiency.

Table 3
Performance comparison of different object detection models on the screwdriver bit dataset.

Model	Precision (%)	Recall (%)	*F*1 (%)	*IoU* (%)	*mAP* (%)
SSD	69.91	71.91	70.51	41.02	71.96
YOLOv7	83.86	88.58	86.15	72.41	96.33
YOLOv7-tiny	54.85	76.76	63.98	46.32	79.52
YOLOv7-CSAM	84.10	87.23	85.64	72.92	95.68
YOLOv4-tiny	74.54	86.66	80.15	63.61	86.6
YOLOv4	93.96	97.24	95.58	82.32	97.69
YOLOv4-CSAM	94.02	97.41	95.69	82.52	97.76

4.3 Angle detection results

In this section, we describe the differences between the actual and estimated angles obtained by Canny- and PCA-based methods. Figure 3 presents the angle detection results for the two compared methods. In Fig. 3, we consider four placement angles (i.e., 45, 90, 135, and 180°), and the experiments for each angle are repeated 30 times. Note that both angles of 0 and 180° stand for a horizontal object. In this study, we used 180° to represent the horizontal angle. As shown in Fig. 6(a), the influence of light causes the angle curve obtained by the Canny-based method to have clear fluctuations, producing an unstable angle output. On the other hand, in the PCA-based method, dimension reduction on the image is first performed to find the feature vectors of the target contour, and then the angle based on the feature direction of the center point is calculated. Figure 6(b) shows that the PCA-based method can generate stable and accurate angle results in 30 tests.

To quantify the test results in Fig. 6, we use *MAE* to calculate the angle estimation error. The *MAE* of the Canny-based method ranges from 0.7 to 2.21, and the *MAE* of the horizontal angle (i.e., 180°) is the smallest among all the angles. Their corresponding values are 0.70 and 0.94, respectively. Compared with the Canny-based method, the *MAE*s of the PCA-based method are all zero among the four test angles. This means that the PCA-based method can provide accurate object angles, allowing the robotic arm to grasp objects with minimal damage.

4.4 Real-time operation results

To verify the screwdriver grasping and placement system proposed in this study, real-time experiments were conducted in two scenarios: static and running conveyor belts. During the experiments, all twelve types of targets were scattered on the conveyor belt, of which six types of screwdriver bits (PH2, PH2L, PH3, PH3L, CIC8, and CIC10) were picking targets, and the remaining six types (1s, 2s, 3s, 7s, PH1s, and PH2s) were regarded as non-picking targets. Figure 7 shows the process of the object placement task: when the system detects that the object on the conveyor belt is the picking target, it calculates the object placement angle, converts the spatial information between the imaging system and the robotic arm system, and controls the robotic arm through ROS to pick up the target and places it at a specific location in the storage box.

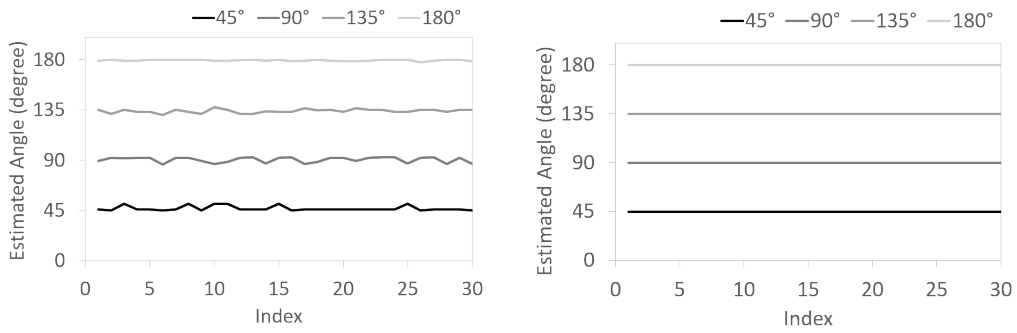

Fig. 6. Angle detection results: (a) Canny- and (b) PCA-based methods.

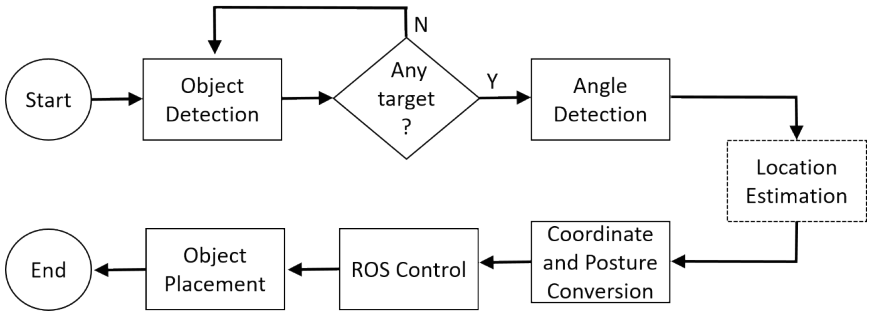

Fig. 7. Object grasping procedure for robotic arms. The dotted block is an additional step under the running conveyor belt scenario.

According to the YOLOv4-CSAM detection sequence for the six picking targets, the system can perform the process shown in Fig. 7 for each target in sequence and finally complete the object placement task. The experiment described above was repeated ten times, and the corresponding statistical results are presented in Table 2. In the static conveyor belt scenario, YOLOv4-CSAM can correctly detect twelve screwdriver bits in 10 tests. For six types of picking targets, 8 out of 10 tests can successfully grasp and place all targets at designated locations, which means that angle detection and subsequent arm control can complete the task. In the 5th and 8th tests, an object placement failure occurs for each test. The overall average object placement success rate is (58/60) × 100% = ~97% accordingly. By inspecting the experiment data, it can be seen that since the screwdriver bits are slender in shape and light in weight, these two failures are attributed to the fact that the distance between the objects is very close, thus causing the grasping error.

In the running conveyor belt scenario, the robotic arm needs to grasp and place objects while the objects are moving. To achieve this goal, in addition to real-time object and angle detection, the robotic arm needs to update the coordinates of the object based on the speed of the object's movement. Accordingly, the process of running the conveyor belt scenario inserts a location estimation stage after angle detection. In the location estimation stage, the object's moving speed and the robotic arm's moving time can be used to estimate the object's position. The robotic arm reaches this estimated position early to perform the object grasping in time. After the location estimation stage, the estimated object's position and angle can be converted into the robotic arm coordinate system for other subsequent stages.

During the experiment in the running scenario, we placed different screwdriver bits on the conveyor belt one by one. The robotic arm would only pick up the objects with the six specified types and skip the rest. Table 4 shows that in ten tests, YOLOv4-CSAM can correctly detect twelve types of screwdriver bits and successfully skip six non-picking objects. For six picking objects, 5 out of 10 tests can complete the placement task, and the successful placement ratio for the remaining five tests ranges from 50 to 83%. The overall success ratio is 87%. The inspection of the experimental data of these placement failures shows that the slight vibration caused by the movement of the conveyor belt causes the position or angle of the lightweight screwdriver bits to shift, ultimately causing the grasping failure.

Table 4
Real-time object placement results for the static and running conveyor belt scenarios.

No.	Successful detection ratio (%)		Successful skipping ratio (%)		Successful placement ratio (%)	
	static	running	static	running	static	running
1	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	100 (6/6)	100 (6/6)
2	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	100 (6/6)	100 (6/6)
3	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	100 (6/6)	83 (5/6)
4	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	100 (6/6)	100 (6/6)
5	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	83 (5/6)	50 (3/6)
6	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	100 (6/6)	83 (5/6)
7	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	100 (6/6)	100 (6/6)
8	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	83 (5/6)	67 (4/6)
9	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	100 (6/6)	100 (6/6)
10	100 (12/12)	100 (12/12)	100 (6/6)	100 (6/6)	100 (6/6)	83 (5/6)
Avg.	100 (120/120)	100 (120/120)	100 (60/60)	100 (60/60)	97 (58/60)	87 (52/60)

4.5 Discussion

In this study, we developed a real-time system to detect and classify screwdriver bits on an assembly line, as well as to control a robotic arm. The overall system architecture, including image acquisition, YOLO-based detection, and robotic arm control integration, was designed with a certain degree of modularity, making it theoretically adaptable to other industrial objects with similar visual characteristics. However, different industrial tasks may exhibit significant differences in object shape, size, reflectivity, and working environment, and further data collection and adjustments to model training strategies and control processes are still necessary.

For real-time use, we predict the object's position on the basis of the conveyor speed and the arm's motion profile, and then send this information directly to the controller. This ensures that the grasping remains on time, even when the conveyor speed changes. The key factor is making sure the robotic arm's speed matches the conveyor speed. The vision-based detection part does not slow down the system because the predicted object's position is used directly.

Our experiments were conducted in a stable, well-lit indoor setting representative of a typical assembly line. In this stage, we focused on building and verifying the core system under controlled conditions.

5. Conclusions

In this study, we aimed to build a robotic arm control system based on deep learning and vision processing methods to perform the object placement task at the assembly line for screwdriver bits. Since screwdriver bits have similar slender shapes, the main difference comes from the tip, which can be regarded as a small object. Therefore, this system employs the YOLOv4-CSAM method to enhance the object detection performance for the screwdriver bits on the conveyor belt. Furthermore, our built system uses PCA to obtain the angle of the slender object and give the gripper accurate posture information to achieve damage-free object grasping. The experimental results showed that (1) YOLOv4-CSAM can achieve an average accuracy of 97.76% for twelve screwdriver bits, which is higher than those of the general YOLOv4 and

YOLOv7 without the attention mechanism; (2) in the static and running conveyor belt scenarios, YOLOv4-CSAM can correctly identify twelve types of screwdriver bits for all tests; and (3) applying PCA to angle detection can obtain a stable object contour, thus achieving 100% angle detection accuracy.

In the real-time experiments, it was observed that the position changes during movement can cause angle detection failure; slight fluctuations in the moving speed of the conveyor belt may also cause the robotic arm to be unable to reach the object in time. One of the future works in this study considers a time-series prediction network to deal with the speed fluctuation problem of the running conveyor belt. For the problem that the position and angle of objects may change with each frame, we can improve YOLOv4 by adding angle information or integrating semantic segmentation methods to effectively track the spatial information of the targets on a frame-by-frame basis.

Future work will aim to utilize this system for various types of objects and other industrial applications. We will gather more data to check if the model works well in new situations. To handle different lighting conditions, we plan to use data augmentation, adjust camera settings (such as auto-exposure or HDR), add extra lighting (such as soft LED or infrared), and test the system under various lighting setups on other production lines. These steps will help make the system more reliable in real-world environments.

Acknowledgments

This research was funded by Taiwan's National Science and Technology Council (NSTC) under Grant No. NSTC 114-2218-E-005-006.

References

1 I. Sobel and G. Feldman: Pattern Classification and Scene Analysis (1973) 271–272. https://www.researchgate.net/publication/285159837_A_33_isotropic_gradient_operator_for_image_processing
2 L. S. Davis: Comput. Graph. Image Process. **4** (2008) 248. https://doi.org/10.1016/0146-664X(75)90012-X
3 M. C. Ergene and A. Durdu: Proc. Int. Artif. Intell. Data Process. Symp. (IDAP) (2017) 1–5. https://doi.org/10.1109/IDAP.2017.8090228
4 R. Girshick: Proc. IEEE Int. Conf. Comput. Vis. (ICCV) (2015) 1440–1448. https://doi.org/10.1109/ICCV.2015.169
5 S. Ren, K. He, R. Girshick, and J. Sun: Proc. Neural Inf. Process. Syst. (NIPS) (2015) 91–99. https://arxiv.org/abs/1506.01497
6 S. Ainetter and F. Fraundorfer: Proc. IEEE Int. Conf. Robot. Autom. (ICRA) (2021) 13452–13458. https://doi.org/10.1109/ICRA48506.2021.9561398
7 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed: Proc. Eur. Conf. Comput. Vis. (ECCV) (2016) 21–37. https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2
8 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) (2016) 779–788. https://doi.org/10.1109/CVPR.2016.91
9 C.-C. Wong, M.-Y. Chien, R.-J. Chen, H. Aoyama, and K.-Y. Wong: IEEE Access **10** (2022) 20159. https://doi.org/10.1109/ACCESS.2022.3151717
10 K. Uçar and H. E. Koçer: IEEE Lat. Am. Trans. **21** (2023) 335. https://doi.org/10.1109/TLA.2023.10015227
11 J. Hu, L. Shen, and G. Sun: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2018). https://doi.org/10.1109/CVPR.2018.00745

12 J. Liu, Z. Wei, Z. Li, X. Mao, J. Wang, Z. Wei, and Q. Zhang: Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) (2024) 7775–7779. https://doi.org/10.1109/ICASSP48485.2024.10447147
13 R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra: Proc. IEEE Int. Conf. Comput. Vis. (ICCV) (2017) 618–626. https://doi.org/10.1109/ICCV.2017.74
14 S. Woo, J. Park, J. Y. Lee, and I. S. Kweon: Proc. Eur. Conf. Comput. Vis. (ECCV) (2018) 3–19. https://doi.org/10.1007/978-3-030-01234-2_1
15 Y. Li, S. Li, H. Du, L. Chen, D. Zhang, and Y. Li: IEEE Access **8** (2020) 227288. https://doi.org/10.1109/ACCESS.2020.3046515
16 M. Li, H. Wang, and Z. Wan: Comput. Electr. Eng. **102** (2022) 108208. https://doi.org/10.1016/j.compeleceng.2022.108208
17 H. Yao, Y. Liu, X. Li, Z. You, Y. Feng, and W. Lu: IEEE Trans. Intell. Transp. Syst. **23** (2022) 22179. https://doi.org/10.1109/TITS.2022.3177210
18 K. Pearson: Philos. Mag. **2** (1901) 559. https://doi.org/10.1080/14786440109462720
19 I. T. Jolliffe and J. Cadima: Philos. Trans. R. Soc. A **374** (2016). https://doi.org/10.1098/rsta.2015.0202
20 Techman Robot (TM5-700): Robot product specification https://uploads.unchainedrobotics.de/media/file_upload/i837_collaborative_robots_datasheet_en_1c831998.pdf (Accessed 28 September 2024).
21 S. Hernandez-Mendez, C. Maldonado-Mendez, A. Marin-Hernandez, H. V. Rios-Figueroa, H. Vazquez-Leal, and E. R. Palacios-Hernandez: Proc. IEEE Int. Autumn Meet. Power Electron. Comput. (ROPEC) (2017) 1–6. https://doi.org/10.1109/ROPEC.2017.8261666
22 C.-Y. Hu, C.-R. Chen, C.-H. Tseng, A. P. Yudha, and C.-H. Kuo: Proc. IEEE Int. Conf. Ind. Technol. (ICIT) (2016) 1632–1636. https://doi.org/10.1109/ICIT.2016.7475006
23 G. Peng, Z. Ren, H. Wang, X. Li, and M. O. Khyam: IEEE Trans. Autom. Sci. Eng. **19** (2022) 3639. https://doi.org/10.1109/TASE.2021.3128639
24 Z. Zhang: IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 1330. https://doi.org/10.1109/34.888718
25 C.-J. Lin, P.-J. Lin, and C.-H. Shih: Sens. Mater. **36** (2024) 1003. https://doi.org/10.18494/SAM4739
26 P. V. C. Hough: Proc. Int. Conf. High Energy Accel. Instrum. (1959). https://inspirehep.net/files/53d80b0393096ba4afe34f5b65152090
27 D. H. Ballard: Pattern Recognit. **13** (1981) 111. https://doi.org/10.1016/0031-3203(81)90009-1
28 A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao: arXiv preprint, arXiv:2004.10934 (2020). http://dx.doi.org/10.48550/arXiv.2004.10934
29 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin: Adv. Neural Inf. Process. Syst. **30** (2017) 5998. https://arxiv.org/abs/1706.03762
30 Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (2020) 11534–11542. https://doi.org/10.1109/CVPR42600.2020.01155
31 A. Hassani, S. Walton, J. Li, S. Li, and H. Shi: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (2023) 6185–6194. https://arxiv.org/abs/2204.07143
32 D. Han, Z. Wang, Z. Xia, Y. Han, Y. Pu, C. Ge, J. Song, S. Song, B. Zheng, and G. Huang: Adv. Neural Inf. Process. Syst. **37** (2024) 127181. https://doi.org/10.48550/arXiv.2405.16605
33 C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao: arXiv preprint, arXiv:2207.02696 (2022). https://arxiv.org/abs/2207.02696
34 N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko: Proc. Eur. Conf. Comput. Vis. (2020) 213–229. https://arxiv.org/abs/2005.12872
35 S. Chen, P. Sun, Y. Song, and P. Luo: Proc. IEEE/CVF Int. Conf. Comput. Vis. (2023) 19830–19843. https://arxiv.org/abs/2211.09788

About the Authors

Cheng-Jian Lin received his B.S. degree in electrical engineering from Ta Tung Institute of Technology, Taipei, Taiwan, R.O.C., in 1986, and his M.S. and Ph.D. degrees in electrical and control engineering from National Chiao-Tung University, Taiwan, R.O.C., in 1991 and 1996, respectively. Currently, he is a chair professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C., and the dean of Intelligence College, National Taichung University of Science and Technology, Taichung, Taiwan, R.O.C. His current research interests are in machine learning, pattern recognition, intelligent control, image processing, intelligent manufacturing, and evolutionary robots. (cjlin@ncut.edu.tw)

Chi-Huang Shih received his Ph.D. degree in electrical engineering from National Cheng Kung University, Taiwan, R.O.C., in 2008. Currently, he is an associate professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C. His current research interests are in machine learning, multimedia network communication, biomedical worn devices, embedded systems, and IoT applications. (chshih@ncut.edu.tw)

Yeong-Yuh Xu received his Ph.D. degree in computer science and information engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 2004. He is currently an associate professor of the Department of Automatic Control Engineering at Feng Chia University, Taichung, Taiwan. His research interests include pattern recognition, neural networks, machine learning, and content-based image and video retrieval. (yyxu@fcu.edu.tw)